



А

ЛЬМАНАХ

НАУЧНЫХ РАБОТ
МОЛОДЫХ
УЧЕНЫХ

2018

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

**АЛЬМАНАХ
НАУЧНЫХ РАБОТ
МОЛОДЫХ УЧЕНЫХ
Университета ИТМО**

Том 7



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург

2018

Альманах научных работ молодых ученых Университета ИТМО. Том 7. – СПб.: Университет ИТМО, 2018. – 337 с.

Издание содержит результаты научных работ молодых ученых, доложенные на XLVII научной и учебно-методической конференции Университета ИТМО по тематикам: экономика; менеджмент; инноватика.

ISBN 978-5-7577-0589-7

ISBN 978-5-7577-0596-5 (Том 7)



УНИВЕРСИТЕТ ИТМО

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, 2018

© Авторы, 2018

ВВЕДЕНИЕ

Издание содержит результаты научных работ молодых ученых, доложенные 30 января – 2 февраля 2018 года на XLVII научной и учебно-методической конференции Университета ИТМО по тематикам: экономика; менеджмент; инноватика.

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.

**Быкова Александра Алексеевна**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра аппаратно-программных комплексов
вычислительной техники, студент группы № S4100Направление подготовки: 09.04.01 – Информатика и вычислительная
техника

e-mail: aleksansha@mail.ru

**Мартынов Владимир Павлович**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра аппаратно-программных комплексов
вычислительной техники, заведующий кафедрой, к.т.н., доцент

e-mail: vpmartynov@corp.ifmo.ru

УДК 555.32

ИНФОРМАЦИОННЫЕ И КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ В СПОРТЕ**Быкова А.А.****Научный руководитель – к.т.н., доцент Мартынов В.П.**

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

В работе проведен анализ развития информационных технологий в индустрии спорта. Систематизированы основные направления внедрения информационных технологий в современный спорт.

Ключевые слова: информационные технологии, коммуникационные технологии, инновации, спорт, спортивные игры.

Актуальность темы исследования. Современный мир все больше связан с информационными технологиями (ИТ). Интеграция компьютеров в нашу повседневную жизнь растет все быстрее и быстрее. Данные изменения коснулись и спорта. На сегодня уже накоплен определенный опыт использования средств ИТ в различных видах спорта, в учебном процессе по физическому воспитанию и в процессе подготовки специалистов (учителей физкультуры, судей по спорту, инструкторов и др.), в процессе спортивных тренировок и соревнований, а также диагностики и мониторинга физического состояния учащихся и спортсменов. Также значительную роль играют программные и аппаратные средства, используемые в обеспечении проведения спортивных соревнований и в индивидуальной практике спортсменов: сверхточная фиксация достижений спортсменов, проведение онлайн-трансляций спортивных мероприятий, устройства для измерения физического состояния спортсменов, программы для регистрации и обработки спортивных результатов. Таким образом, внедрение ИТ в спорт является требованием времени. Всем вышеупомянутым обуславливается актуальность данной работы.

Цель работы – анализ основных направлений применения и перспектив развития информационных технологий в спорте.

На сегодняшний день ИТ широко используется в самом спорте и разных сферах, которые окружают современный спорт. Рассмотрим данные направления.

Так, на сегодняшний день уже невозможно представить образовательный процесс без использования электронных пособий для теоретической подготовки по отдельным видам спорта, что позволяет в дальнейшем осмысленно и продуктивно применять полученные знания на практике.

Одним из самых популярных продуктов ИТ в спорте сегодня является фитнес-браслет. Это устройство подключает программное обеспечение на вашем смартфоне и обеспечивает двустороннее соединение с вашим телом. Он отправляет сигналы о пульсе, учитывает количество шагов в день, калории и другие характеристики в зависимости от модели и производителя. Вся информация отображается в приложении или даже на браслете.

О.П. Кокоулина [1] указывает, что при проведении соревнований уровня олимпийских игр применение ИТ обеспечивает оперативный сбор, передачу, хранение и обработку большого количества информации. Речь идет о фото- и видеозаписи, спортивные прогнозы и т.д. Высококачественный видеозахват позволяет выполнять покадровое замедление, которое, в свою очередь, используется для мониторинга и обзоров некоторых игровых моментов. Это помогает определить спортсмена, который пришел первым, если возникает спорная ситуация, или найти причину чьей-то ошибки, или воспроизвести специальный момент. В таких видах спорта как теннис и футбол используется система электронного судейства Hawk-Eye. Видеокамеры высокого разрешения располагаются вокруг игровой зоны (корт или поле), отслеживают траекторию полета мяча и фиксируют информацию, которая хранится в системе электронного судейства. Это позволяет в спорных ситуациях для вынесения решения наблюдать на экране компьютера и на больших табло движение мяча, точку его приземления и так далее [2].

Еще одно направление развития ИТ в спорте обеспечивают программные продукты, вида Smart Soccer Ball. Это устройство помогает игрокам в футбол отслеживать свои навыки в цифровом виде. Сам мяч имеет датчик для измерения всех основных характеристик: скорости шара, вращения, траектории и самой важной вещи – удара. Датчик расположен в центре мяча, собирает все данные и отправляет их на сопряженное беспроводное принимающее устройство. Игрок знает все детали игры, а также имеет возможность анализировать особенности игры других футболистов. Мяч официально был утвержден ФИФА, и игроки не чувствуют разницы играя с мячом Smart Soccer Ball, так как он такого же веса и формы [1].

Другим уникальным изобретением, которое помогает изучить игровые приемы, является баскетбольная площадка в Шанхае, так называемый «Дом Мамбы». Она разработана АКQA и Nike, и позволяет отображать рекомендуемое направление движения во время игры. Пол площадки снабжен датчиками, которые подают световые сигналы и моделируют тактику игры. Всем комплексом управляют профессионально разработанные учебные программы, поэтому тренеры могут загружать программы для своих команд. Искусственный интеллект помогает игрокам практиковать навыки, необходимые для игры, а мышечная память формируется так, чтобы спортсмены знали, что делать, когда они играют по-настоящему.

Использование беспроводного фиксатора уколов для фехтования «Идея» позволяет фехтовальщикам отказаться от длинных шнуров. Информация об уколе поступает на датчик, расположенный на затылочных держателях маски, и передается на судейский приемник через инфракрасный порт. Одновременно внутри маски загорается красный или зеленый светодиодный индикатор, освобождает судей, зрителей и спортсменов от необходимости обращаться к центральному судейскому регистратору. Кроме того, фехтовальщики получают возможность вести бой не только на специальной дорожке, но и на любой другой площадке.

Композитный мяч для баскетбола Spalding's Cross Traxxion изготовлен из композитного синтетического материала, который технологичнее, чем прежние кожаные покрытия. Композитный мяч хорошо сбалансирован, его поверхность позволяет контролировать уровень влажности, и избавляет от необходимости его обкатки.

Разработанны коньки новой конструкции Therma Blade с горячими лезвиями, батареей и микропроцессором. При постоянном нагреве лезвия конька (на несколько градусов выше температуры льда) увеличивается прослойка воды под лезвием конька, и уменьшается трение между ним и льдом, за счет чего увеличивается скорость прохождения дистанции, улучшается скольжение [3].

Изменения в правилах спортивной гимнастике, новые технологии подготовки спортсменов диктуют необходимость систематического повышения квалификации судей, тренеров и спортсменов. Для решения этой проблемы создаются компьютерные программы по правилам соревнований в спортивной гимнастике на основе универсальной информационно-диагностической системы [4].

Особого внимания заслуживают технологии, которые являются частью спортивной формы и экипировки. Умная футбольная форма Under Armour E39, которая тоже является разработкой компании Adidas, оснащена датчиками, которые измеряют важнейшие процессы жизнедеятельности организма спортсмена. Эта технология позволяет прогнозировать действия игрока во время футбольного матча или на тренировке, исходя из показаний датчика о его физическом состоянии. Процесс передачи информации тренеру заключается в следующем: встроенные сенсоры в режиме онлайн измеряют давление, температуру тела, параметры дыхания футболиста. Данная информация записывается на жесткий диск встроенного датчика и выполняет передачу информации на компьютер. Тренер имеет возможность воспользоваться этой технологией во время игры с помощью iPad или iPhone.

На сегодняшний день в России особой популярностью пользуется биатлон. Для многих любителей биатлона получение информации о результатах стрельбы, промежуточном и итоговом времени участников биатлонной гонки – само собой разумеющееся дело. Присутствуя на игре, мы можем наблюдать, как данные соревнований немедленно появляются на информационных табло и экранах телевизоров, и всегда точно отражена ситуация. Однако все не так просто. В течение многих лет Кристиан Винклер из фирмы SIWIDATA и его команда работали над разработкой и продолжают постоянно совершенствовать свою систему хронометража, которая только на первый взгляд кажется довольно простой. На самом деле по информации, которую видят зрители, стоят серьезные технологии. Важнейший элемент системы – транспондеры, т.е. датчики красного цвета, закрепленные по одному на каждой ноге спортсмена [5].

Камера фотофиниша при необходимости дает еще более глубокую детализацию. В спринте и индивидуальной гонке на финише кроме камеры используются еще и фотоэлементы, которые обеспечивают точный результат. Сигнал с транспондеров на трассе снимают не менее десяти ресиверов. Каждый представляет собой отрезок трубы, начиненный электроникой, и соединенный с передатчиком, который транслирует информацию через беспроводной модем или кабель. Вся информация автоматически стекается в хронометражную комнату. В индивидуальных гонках от каждого спортсмена с трассы, стрельбища и финиша поступает около 40 пакетов данных. Обязательное условие – полное отсутствие сбоев. Из всех поступающих данных моментально снимается резервная копия, а для телевизионной графики предусмотрено тройное резервирование. Единственная операция, которая выполняется вручную, – введение в систему номера спортсмена, когда он приходит на установку для стрельбы. С помощью беспроводной связи данные в режиме реального времени передаются в несколько мест сразу. Важнейшее из них – телевизионные аппаратные, где формируется экранная графика. Следующие по приоритету – комментаторы и тренеры. На дистанции гонки оборудованы три зоны, где специалисты из команд могут принимать сигнал и получать на компьютеры всю информацию о гонке. Одновременно те же данные поступают в Интернет для общего пользования.

Программное обеспечение спортивного прогнозирования набирает популярность, оно предлагает статистику и историю для прогнозирования победителей следующей игры, и сделать ставку. Это поможет любителям спорта зарабатывать деньги на своих увлечениях.

Ученые работают над технологиями генетических алгоритмов, чтобы сделать прогнозы более точными и реальными [6].

Выводы. Проведенный анализ показывает, что за последние годы информатизация современного общества приобрела новые масштабы. Проведение спортивных соревнований с использованием ИТ является более качественным и эффективным. На сегодня ИТ нашли довольно широкий спектр применения в спорте и используются:

1. для информационно-методического обеспечения и управления организационным и учебно-тренировочным процессом в спортивных организациях;
2. для создания моделей тренировочных и соревновательных ситуаций;
3. при организации мониторинга физического состояния и здоровья спортсменов;
4. как средство автоматизации процессов обработки результатов соревнований;
5. как средство автоматизации процессов контроля, компьютерного тестирования физического, функционального, умственного и психологического состояний спортсменов и коррекции результатов учебно-тренировочной деятельности;
6. также ИТ активно внедряются в рекламную и предпринимательскую деятельности в сфере спорта.

Итак, на сегодня спортивная жизнь зависит не только от физических достижений человека, но и от изобретений инженеров, разработчиков, химиков и тех, которые пытаются использовать науку и информацию технологий для улучшения результатов действий спортсмена. Важно и то, что информационные технологии способствуют популяризации спорта среди широких слоев населения.

Литература

1. Кокоулина О.П. Информационные технологии в физической культуре и спорте // Приоритетные научные направления: от теории к практике. – 2017. – С. 235–245.
2. Сайт компании Hawk-eye Innovations [Электронный ресурс]. – Режим доступа: <https://www.hawkeyeinnovations.com/>, своб.
3. 10 Techs Transforming Sports [Электронный ресурс]. – Режим доступа: <https://www.seeker.com/10-techs-transforming-sports-1766272950.html>, своб.
4. Белоусова Д.А. Информационные технологии в системе «физическая культура и спорт» [Электронный ресурс]. – Режим доступа: <https://files.scienceforum.ru/pdf/2015/14928.pdf>, своб.
5. Спорт высоких инноваций. ТОП-10 лучших примеров слияния спорта и технологий. [Электронный ресурс]. – Режим доступа: <http://www.novate.ru/blogs/140813/23740/>, своб.
6. Информационные технологии, которые могут изменить современный футбол // Молодежный научный форум: Технические и математические науки: электр. сб. ст. по материалам XIX студ. междунар. заочной науч.-практ. конф. – 2014. – № 12(19). – С. 15–17.

**Волков Виктор Олегович**

Год рождения: 1992

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра аппаратно-программных комплексов
вычислительной техники, студент группы № S4100Направление подготовки: 09.04.01 – Информатика и вычислительная
техника

e-mail: volkof92@mail.ru

УДК 004.77**ЛОКАЛЬНАЯ ВЫЧИСЛИТЕЛЬНАЯ СЕТЬ 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 работе изложено предназначение мобильного узла связи, раскрыты его возможности, и представлена схема и оборудование, входящее в состав его сети.

Ключевые слова: мобильный узел связи, чрезвычайная ситуация, локальная вычислительная сеть, Интернет, Интранет.

К сожалению, в современном мире все чаще и чаще происходят чрезвычайные ситуации (ЧС) природного и техногенного характера, которые могут иметь большие масштабы как по количеству привлеченных сил и средств к их ликвидации, так и по охвату значительной территории, например, такие как: крупные наводнения, землетрясения, техногенные аварии и т.п. [1, 2].

Для эффективного управления силами и средствами спасательных формирований при ликвидации последствий ЧС используются средства связи, обеспечивающие надежную передачу оперативной информации из района ЧС в пункты управления.

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.

Мобильный узел связи предназначен для обеспечения устойчивой связью в полевых условиях оперативных групп подразделений МЧС России при ликвидации последствий чрезвычайных ситуаций природного и техногенного характера, включая:

- 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.

Оборудование связи и передачи данных позволяет полноценно внедрить МУС в ведомственную цифровую сеть с интеграцией услуг (ВЦССИУ МЧС России), предназначенную для передачи различных видов информации (телефонной, факсимильной, аудио- и видеоконференцсвязи).

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 размещается 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, пользоваться средствами радиосвязи (рис. 1, а).



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 видеоконференцсвязи (ВКС);
- система видеонаблюдения;
- средства визуализации получаемой информации;
- оборудование приема спутникового телевидения и воспроизведения медиаконтента;
- навигационно-связное оборудование;
- система электропитания;
- система 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.

Центрами данных «звезд» являются коммутаторы Cisco 2960, которые имеют поддержку технологии PoE, использующуюся для питания IP-телефонов Cisco.

По периферии внутренней сети Интранет располагаются такие устройства, как ноутбуки, IP-телефоны, МФУ, терминал ВКС, выносные камеры для мониторинга окружающей обстановки, коммутатор видеопотоков, коммутатор радиосредств,

позволяющий соединять средства радиосвязи (радиостанции КВ и УКВ диапазона) с цифровой IP-телефонией.

В подсети пользователей Интернет, помимо ноутбуков на рабочих местах, также имеется Wi-Fi-роутер, позволяющий подключаться к сети Интернет по беспроводному каналу.

Особый интерес представляет подсеть ввода. 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 80 см (рис. 2).



Рис. 2. Мобильный узел с развернутой спутниковой станцией

Дополнительно в комплект МУС входит узел привязки, представляющий собой защитный кофр в котором находится коммутатор Cisco, DSL-модем, интернет-центр и питание для радиорелейной станции «точка-точка». Узел привязки (УП) позволяет привязываться к сети Интернет у доступного оператора связи и передавать Интернет в МУС по одному из имеющихся каналов (рис. 3).

Схема ЛВС мобильного узла связи

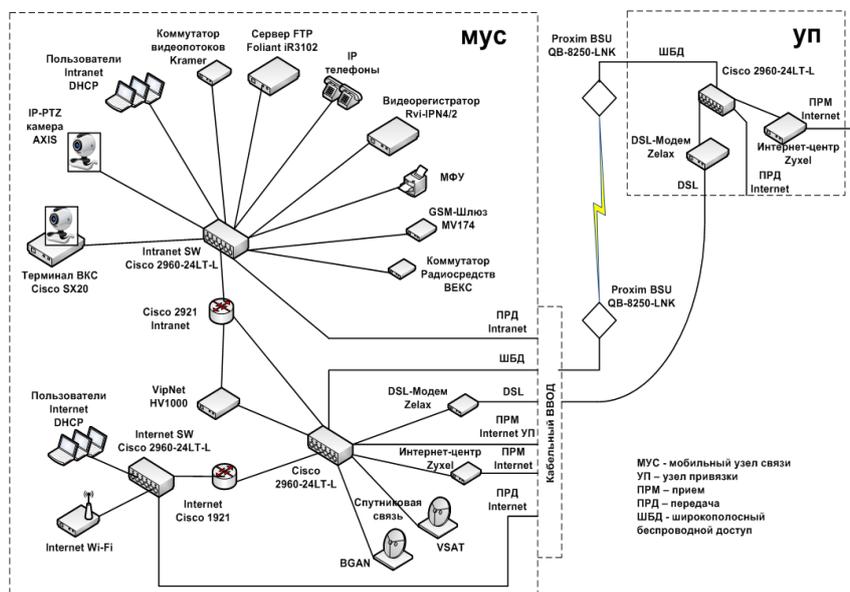


Рис. 3. Схема мобильного узла связи

В данной работе было рассмотрено предназначение мобильного узла связи, его возможности, конструктивное деление на отсеки, а также его ЛВС. В дополнение составлена схема его ЛВС с расположением основных коммутирующих и периферийных устройств.

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.



Глебов Родион Геннадьевич

Год рождения: 1981

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра аппаратно-программных комплексов
и вычислительной техники, студент группы № S4200

Направление подготовки: 09.04.01 – Информатика и вычислительная
техника

e-mail: thomas_cp92@mail.ru

УДК 004.3

**ИССЛЕДОВАНИЕ МЕТОДОВ И СРЕДСТВ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ
В КОРПОРАТИВНЫХ СЕТЯХ НА ПРИМЕРЕ МОБИЛЬНЫХ УСТРОЙСТВ
С ИСПОЛЬЗОВАНИЕМ МОБИЛЬНОГО КОМПЛЕКСА ViPNet**

Глебов Р.Г. (Университет ИТМО)

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

(Российский государственный педагогический университет им. А.И. Герцена)

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

В работе проведено исследование основных процессов работы в защищенных корпоративных сетях на основе мобильного программного комплекса ViPNet – построенного на основе VPN-соединений, проведено сравнение данного мобильного комплекса с другими мобильными приложениями на основе VPN-соединений, выделены основные преимущества мобильного программного комплекса ViPNet для защиты информации в корпоративных сетях.

Ключевые слова: исследование, VPN, ViPNet, защищенные сети, защита информации, комплекс, администратор, клиент, координатор.

В данной работе рассмотрены эффективные средства создания защищенной и отказоустойчивой корпоративной компьютерной сети на основе мобильного программного комплекса ViPNet. Приведены основные его преимущества.

В данной работе предметом исследования выбран программный комплекс ViPNet, так как он имеет ряд особенностей, ставящих его на особое место. Хотя в настоящее время похожие продукты предлагают многие компании: как отечественные, так и иностранные. Например, у компании «Код Безопасности» есть даже встроенное решение на базе планшета, а также отдельный продукт Secret MDM и решения CheckPoint и StoneSoft. Но последние не имеют действующих сертификатов ФСБ России на соответствие требованиям к средствам криптографической защиты информации (СКЗИ). В этой связи целесообразно рассмотреть и оценить решения на «ИнфоТекс» и «Код Безопасности».

Используя открытые источники, выяснилось что решение от «Кода Безопасности» не всегда корректно работает или имеет сложности при работе через различные мобильные устройства, использующие медленные каналы связи (GRPS, EDGE) и каналы с большой задержкой (спутниковые) [1].

А в настоящее время много удаленных точек при командировках сотрудников не имеют хорошего и устойчивого покрытия мобильной сети. И поэтому был проведен обзор именно продукта компании «ИнфоТекс».

Особенностью комплекса является то, что он имеет сертификаты ФСБ России, а также тот факт, что он является одним наиболее используемых коммерческими и государственными структурами, которые используют информацию ограниченного доступа. Также в данном комплексе есть поддержка мобильных устройств. По информации с

информационного ресурса в самых популярных мобильных устройствах (Samsung, YotaDevice(YotaPhone)) данный комплекс уже интегрирован [2].

Еще одной особенностью данного комплекса можно выделить то, что он входит в реестр российского программного обеспечения (ПО), созданного в рамках мер, предпринимаемых государством с целью ограничить закупки импортного ПО в госструктурах и компаниях с государственным участием (реестр заработал с 01.01.2016).

Еще одним преимуществом является возможность заказчику не перестраивать работающую инфраструктуру в конце 2018 года, для соответствия новым требованиям ФСБ России. Так как приложение ViPNet Client for Android имеет действующий сертификат ФСБ России на соответствие требованиям к СКЗИ класса КС1.

Комплекс обеспечивает удаленное защищенное подключение к корпоративным ресурсам используя мобильные устройства с операционной системы (ОС) Android и с ОС iOS.

Приложение позволяет блокировать прямое неконтролируемое обращение с мобильных устройств в открытую сеть Интернет и позволяет организовать доступ к необходимым ресурсам сети Интернет с использованием корпоративных прокси-серверов и межсетевых экранов и корпоративных антивирусов. Это достигается посредством создания VPN-туннеля между корпоративными серверами безопасности и мобильными устройствами и блокирование любого несанкционированного открытого входящего и исходящего трафика с устройства. В результате может быть обеспечена многоуровневая защита информационного обмена мобильных пользователей от вредоносного ПО и компьютерных атак, подчиненная общей политике информационной безопасности компании [3].

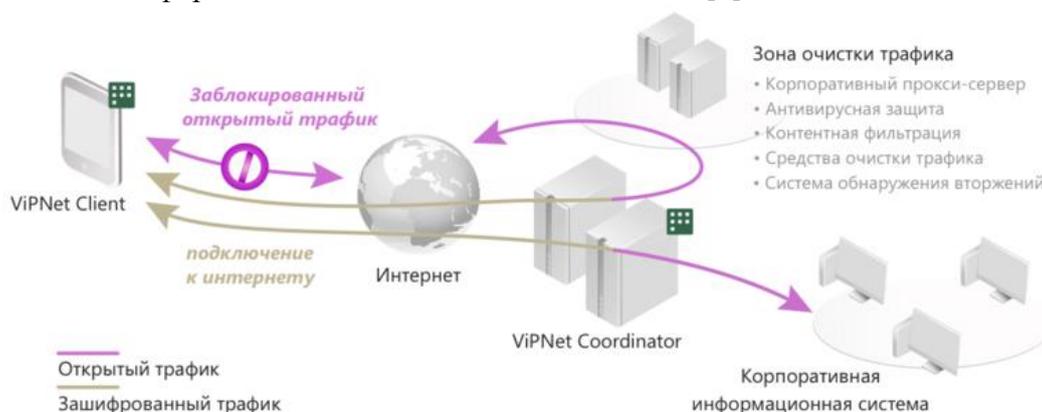


Рис. 1. Организация обмена информацией

Иллюстрация порядка организации обмена информации приведена на рис. 1. Из рисунка видно, что основным преимуществом данной модели построения обмена трафиком является централизованное управление политиками информационной безопасности: антивирусная защита, прокси-сервер и др. Благодаря этому для мобильного устройства обеспечиваются такие параметры защиты:

- защита iPad, iPhone и мобильных устройств с ОС Android от сетевых атак;
- доступ посредством защищенных технологиями ViPNet VPN-туннеля к защищенным ресурсам сети;
- перехват любого IP-трафика, обеспечивая его прозрачное шифрование [1]. Данное ПО обеспечивает эффективную многоуровневую защиту мобильного устройства – антивирусную защиту и контентную фильтрацию, причем без установки дополнительного ПО на каждое мобильное устройство, что немаловажно, учитывая ограниченные возможности автономной работы мобильных устройств.

При демонстрации работы ViPNet на мобильном устройстве следует отметить, как используются различные конфигурации. Конфигурация ViPNet Client для мобильных

приложений – это фиксированный набор параметров работы приложения, предназначенных для настройки параметров доступа к корпоративным ресурсам и ресурсам Интернета.

Мобильные приложения ViPNet используют следующие конфигурации:

- заблокировать сеть – блокировка всех соединений;
- отключить защиту – отключение обработки IP-трафика (соединение с защищенными ресурсами невозможно, доступ к ресурсам Интернет разрешен, но при этом защита IP-трафика нет);
- VPN и Интернет – доступ к защищенным ресурсам и ресурсам Интернета (открытый трафик передается или через корпоративный прокси-сервер, или через координатор, защищенный трафик передается через координатор):
- прямой доступ;
- шлюзовой координатор;
- корпоративный прокси-сервер (предусмотрена только для мобильного приложения ViPNet Client for iOS).

На рис. 2 представлен снимок экрана телефона Samsung, иллюстрирующий работу приложения.

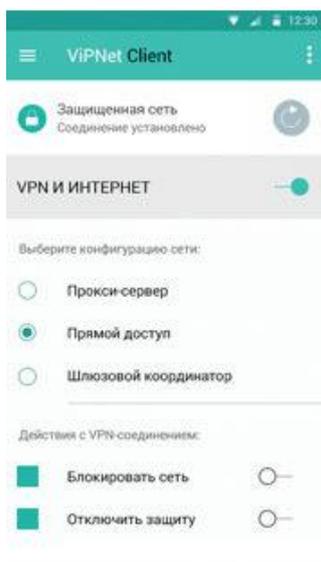


Рис. 2. Выбор режима работы приложения

Остановимся более подробно на устройстве и назначении основных режимов работы:

1. прямой доступ. При выборе данной конфигурации возможна работа с ресурсами защищенной сети ViPNet и прямой неограниченный доступ к открытым ресурсам Интернета. При работе в данной конфигурации отсутствует возможность контроля открытого трафика абонентского пункта, развернутого на устройстве.

Использовать данную конфигурацию целесообразно для работы в тех случаях, когда есть уверенность в том, что на устройстве отсутствует информация, к которой нежелателен доступ посторонних лиц;

2. шлюзовой координатор. При выборе данной конфигурации работа с ресурсами защищенной сети ViPNet и доступ к открытым ресурсам Интернета осуществляется через координатор, выполняющий роль сервера IP-адресов. При работе в данной конфигурации контролируется весь трафик абонентского пункта, развернутого на устройстве.

Использовать данную конфигурацию можно для работы, например, находясь в кафе или в аэропорту, предоставляющем мобильный доступ к ресурсам сети Интернет (Wi-Fi и 3G). Так как все соединения с узлами защищенной и открытой сетей осуществляются через координатор, то для администратора точки доступа они будут невидимы, что исключает возможность перехвата трафика;

3. корпоративный прокси-сервер. При выборе данной конфигурации все соединения с открытыми узлами осуществляются через защищенный прокси-сервер. Подключения к защищенным узлам ViPNet по протоколу HTTP требуется разрешить на прокси-сервере, подключение к защищенным узлам по другим протоколам не ограничено.

Соединение с прокси-сервером осуществляется по защищенному каналу, при этом на прокси-сервере осуществляется обработка трафика в соответствии с корпоративными политиками безопасности (например, защита от вирусов и сетевых атак).

Используют данную конфигурацию, например, при поездке в командировку, в которую необходимо взять с собой мобильное устройство для возможности обмена с коллегами различными служебными данными (например, презентациями, документами, электронными таблицами). При работе в данной конфигурации вся передаваемая информация будет защищена от несанкционированного доступа.

Набор конфигураций, доступный пользователю, зависит от уровня его полномочий, который задает администратор сети ViPNet [2, 3].

В данной работе автором были рассмотрены основные режимы работы приложения, которое является частью аппаратно-программного комплекса ViPNet.

Проведя анализ и обзор функций, был сделан вывод, что отличительной особенностью ViPNet Client для мобильных устройств являются: возможность работы через любые каналы связи, прозрачное переключение между ними, автозагрузка приложения, запрет удаления приложения с мобильного устройства, а также автоматическое восстановление VPN-канала после утери связи. Эта особенность позволяет сотруднику не следить за подключением к защищенной сети и спокойно работать с корпоративными ресурсами [4–7].

Литература

1. Официальный сайт компании Landata [Электронный ресурс]. – Режим доступа: ftp://ftp.landata.ru/Huawei_Symantec/Comparison/Comparison_VPN_2010_v.6.ppt (дата обращения: 26.12.2017).
2. Официальный сайт «Аналитический центр Anti-Malware.ru» [Электронный ресурс]. – Режим доступа: anti-alware.ru/reviews/ViPNet_Client_for_Android_and_ViPNet_Connect (дата обращения: 12.10.2017).
3. Официальный сайт компании Инфотекс [Электронный ресурс]. – Режим доступа: infotecs.ru (дата обращения: 12.10.2017).
4. Гусев В.В. Программно-аппаратные комплексы ViPNet. – М.: Горячая линия – Телеком, 2014. – 354 с.
5. Гусев В.В. Администрирование системы защиты информации ViPNet. – М.: Горячая линия – Телеком, 2013. – 379 с.
6. Кабакова Н.В., Чефранова А.О., Алабина Ю.Ф. Система защиты информации ViPNet. Курс лекций. – М.: ДМК-Пресс, 2015. – 456 с.
7. Чефранова А.О. Технология построения VPN ViPNet. – М.: Горячая линия – Телеком, 2009. – 28 с.

**Кожин Кирилл Романович**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра аппаратно-программных комплексов
вычислительной техники, студент группы № S4100Направление подготовки: 09.04.01 – Информатика и вычислительная
техника

e-mail: kirgodianx@mail.ru

**Костеж Владимир Александрович**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра аппаратно-программных комплексов
вычислительной техники, к.т.н., доцент

e-mail: k@limtu.ru

УДК 004.657

**АНАЛИЗ СИСТЕМ УПРАВЛЕНИЯ КОНТЕНТОМ ДЛЯ РАЗРАБОТКИ
И ПРОДВИЖЕНИЯ WEB-САЙТОВ****Кожин К.Р.****Научный руководитель – к.т.н., доцент Костеж В.А.**

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

В работе выбрана главная цель: проанализировать системы управления (CMS) с которыми автору приходилось работать, и выделить из них основные плюсы и минусы. Работа являлась подготовительным шагом к написанию CMS, которой стали бы присущи основные плюсы других систем направления. Для автора это будет достаточно большим шагом в развитии себя, как WEB-разработчика.

Ключевые слова: CMS – Content management system (система управления контентом), WordPress – CMS, OpenCart – CMS, Bitrix – CMS.

Введение. Сейчас автор работает в небольшой web-студии, и каждый его день состоит из правок, дописания и создания нового функционала для разных сайтов, у каждого из которых есть какая-нибудь система управления. Самые частые CMS (Content Management System), которые встречаются когда-либо это: Wordpress (самая часто встречаемая система управления, в ходе работы стало понятно, почему это так), Opencart. 1С Bitrix, и разные самописные [1–4]. По мнению автора, которое сложилось благодаря работе в web-программировании, самой основной задачей CMS является простота и удобство администрирования сайта, который основан на ней. В это входит:

1. создание страниц;
2. наполнение страниц;
3. подключение модулей;
4. отключение модулей;
5. удаление модулей;
6. добавление медиафайлов;
7. изменение шаблонов;
8. доступ к файлам сайта напрямую из самой системы управления;

9. понятность не программисту, а обычному пользователю (заказчику).

Автор считает, что каждая система управления контентом должна отвечать этим простым по смыслу, но не простым в реализации пунктам. Зачастую системы управления пишутся большой командой разработчиков, которые стараются внести в нее как можно больше различного функционала и совершенно забывают про дружественный интерфейс. В итоге появляется множество вкладок, различных подменю, из-за которых не то что заказчик запутается, но и программист, который начнет работать с ней, первое время будет не понимать, где что подключить и дописывать. Хотя зачастую такое происходит именно в платных CMS на подобие 1С Bitrix.

Методы исследования. Работа автора основана на создании и дописании различных сайтов, поэтому метод исследования основан на практике использования разных систем управления контентом. В это входит:

1. создание сайта с нуля;
2. доработка сайта;
3. оптимизация сайта;
4. написание нового функционала без использования уже готовых модулей;
5. использование готовых модулей для решения конкретных задач.

Исходя из этих пунктов, и будут выделяться плюсы и минусы систем управления, с которыми автору приходилось и приходится работать.

Результаты. Оценка данных CMS приведена в таблице.

Таблица. Оценка данных CMS на основе опыта работы автора

Название	Подключение модулей	Удобство в модификации своим кодом	Удобство администрирования	Итог:
WordPress	8	10	10	28/30
OpenCart	5	8	9	22/30
1С-Bitrix	9	5	6	20/30

Обсуждение. CMS – система управления содержимым (контентом) – компьютерная программа или информационная система, которая используется для организации и обеспечения процесса по совместному созданию, управлению и редактированию содержимого сайта.

Если смотреть с точки зрения обычного заказчика, то разработка сайта на основе какой-либо CMS должна приносить следующие преимущества:

- в работе используется наиболее эффективный инструмент для решения конкретной задачи (в зависимости от вида сайта и требований к его функционалу подбирают оптимальную CMS);
- использование CMS позволяет владельцу сайта самостоятельно создавать и удалять разделы сайта, редактировать различную информацию без привлечения стороннего специалиста – это одно из преимуществ над статическими сайтами;
- работа сайта постоянно тестируется множеством пользователей, а найденные ошибки и уязвимости достаточно оперативно устраняются, при этом сайт работает на самых передовых и проверенных технических решениях;
- временные затраты на разработку сайта существенно снижаются, так как разработчику не надо фиксировать свое внимание на чисто технических задачах: «как сделать ленту с новостями» или «как научить CMS искать товары в каталоге», а можно сосредоточиться на информационной и визуальной составляющих будущего сайта.

1. WordPress. CMS WordPress имеет просто огромный набор готовых шаблонов – это варианты будущего сайта, которые при небольшом знании PHP можно кардинально

менять под свои потребности. А без знания программирования можно управлять рубриками, создавать новые страницы и записи, вставлять в них картинки, таблицы, видео и т.д., добавлять или убирать виджеты.

Но WordPress предназначен не только для любителей. Этой системой пользуются и профессиональные веб-дизайнеры и программисты. WordPress помогает не наполнять каждый сайт с нуля множеством скриптов и новыми стилями разметки – можно использовать уже существующие варианты как точку отсчета для собственной разработки. Сайт на WordPress – это не только функционал для упрощенной разработки сайта – это целая система программ, которая работает на сайте постоянно, поддерживая его жизнедеятельность – от начала до конца. Такая система и называется CMS, т.е. система управления содержимым сайта.

WordPress очень удобен в создании сайта, так как он имеет не только дружелюбный интерфейс, но и удобную файловую структуру, благодаря которой проще искать нужные папки и файлы конфигураций, чтобы вносить свои изменения не только в работу сайта, но и в саму систему управления. К сожалению WordPress не во всем хорош, если большинство функционала добавляется с помощью плагинов (модулей), то с очень большой вероятностью, через какое-то время начнутся конфликты между ними и работоспособность сайта упадет. Но это встречается реже, чем в другой системе управления OpenCart.

2. OpenCart. OpenCart – CMS с открытым кодом, имеющая обширное сообщество пользователей, назначением которой является создание площадок для электронной коммерции. Этот движок немного проигрывает другим системам в плане медийной раскрученности, но справляется с поставленными задачами не хуже некоторых коммерческих CMS, поэтому однозначно заслуживает внимания.

Преимущества:

- наличие активного русскоязычного форума;
- высокая скорость работы и небольшая нагрузка на сервер в сравнении с другими CMS (той же 1С-Битрикс) для создания интернет-магазинов;
- собственная система формирования отчетов.

Если на форуме нет решения какой-то проблемы, то можно задать вопрос и получить адекватный ответ от более опытных пользователей. Системой OpenCart пользуются люди с разным уровнем компьютерной грамотности, поэтому даже если вы раньше не делали ничего сложнее переустановки Windows, то благодаря поддержке сообщества веб-мастеров быстро научитесь работать с движком и управлять сайтом.

Со стороны клиентов OpenCart предлагает множество интересных возможностей, облегчающих посетителям поиск и покупку в интернет-магазине.

Недостатки:

- проблемы с индексацией удаленных страниц;
- ошибки при формировании адресов;
- отсутствие контроля дубля файлов (если у картинок одинаковые имена, то новый файл просто затирает предыдущее изображение).

У сайтов, которые работают на OpenCart, есть определенные проблемы с SEO-оптимизацией. Без специальных расширений возникают трудности с отображением кода ошибки 404. Удаленные страницы продолжают индексироваться поисковыми роботами, что приводит к снижению места сайта в поисковой выдаче. Кроме того, адреса формируются таким образом, что одна страница может открываться по разным URL.

Главной проблемой CMS OpenCart является совместимость версий. Допустим, вы хотите поставить на тот же самый WordPress старую тему, со старой версии движка на новую. Вам достаточно нажать одну кнопку и все заработает. OpenCart же просто откажется работать. Точно также происходит со всеми плагинами. Нужно внимательно следить, чтобы версии плагинов и темы совпадали с версией системы управления иначе можно потерять весь сайт.

3. CMS 1С-Битрикс. CMS 1С-Битрикс – это автоматизированная система управления контентом, разработанная компанией 1С-Битрикс. Продукт предназначен для создания и развития корпоративных проектов предприятий и организаций, информационных, новостных и справочных порталов, социальных сетей, интернет-магазинов и других видов сайтов.

Основное преимущество системы CMS от 1С-Битрикс заключается в отсутствии необходимости знания языка гипертекстовой разметки и языков веб-программирования. Следует только выполнить предварительную настройку продукта – создать шаблоны дизайна, структуру разделов и интернет-страниц, а также подключить ряд дополнительных модулей. Реализацию данных задач берут на себя партнеры компании.

Многосайтовость. Разработчики CMS 1С-Битрикс считают, что продукт может эффективно решать задачи 95% проектов. Исключение составляют только те сайты, которые имеют очень индивидуальный характер и созданы на основе специфической логики.

Полноценный визуальный редактор. Визуальный редактор снабжен функцией проверки орфографии и позволят удачно осуществлять вставку объектов и текста из MS Word. Посредством редактора очень легко изменять текст на страницах, у ссылок, в таблицах и работать с динамичными элементами сайта.

Интеграция с программой 1С:Предприятие. CMS 1С-Битрикс – это система, созданная в результате взаимодействия компаний Битрикс и 1С, поэтому его пользователю обеспечена уверенная интеграция с продуктом 1С:Предприятие, что весьма удобно, допустим для фирм, которые занимаются реализацией различных товаров и услуг в Интернете.

Инструменты для разработки индивидуальных решений. При условии соответствующих знаний платформа Frame Work позволяет внести значительные изменения в оболочке CMS 1С-Битрикс и в результате создать собственную систему управления веб-проектами.

Магазин MarketPlace. На сайте магазина MarketPlace для пользователей 1С-Битрикс: Управление сайтом партнерами фирмы предлагаются всевозможные бесплатные и платные модули, позволяющие существенно расширить функциональность проекта.

Заключение. При работе с данными системами управления определено то, как разработчики реализуют в своих системах примерно одинаковый функционал только разными способами. Как устроен интерфейс, как реализовано создание страниц и структуры баз данных CMS.

Литература

1. Как сделать модуль 1С-Битрикс [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/196376/>, своб.
2. CMS OpenCart [Электронный ресурс]. – Режим доступа: <https://opencart-cms.ru/opencart/>, своб.
3. Несколько слов о движке интернет-магазинов OpenCart [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/225081/>, своб.
4. Что такое WordPress [Электронный ресурс]. – Режим доступа: <https://max1net.com/cms-wordpress-osnovy-preimushhestva-i-rabota-po-organizacii-sajta/>, своб.

**Кудрявых Антон Юрьевич**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра аппаратно-программных комплексов
вычислительной техники, студент группы № S4100Направление подготовки: 09.04.01 – Информатика и вычислительная
техника

e-mail: dart-shaul@mail.ru

УДК 004.49**ОСОБЕННОСТИ ПРИМЕНЕНИЯ СРЕДСТВ СЕТЕВОГО АНАЛИЗА И ЗАЩИТА
ОТ НИХ****Кудрявых А.Ю.****Научный руководитель – к.т.н., доцент Мартынов В.П.**

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

В работе были исследованы особенности применения средств сетевого анализа, рассмотрены сетевые мониторы, анализаторы трафика (снифферы), особенности их применения, преимущества и недостатки. Рассмотрены основные опасности в использовании анализаторов трафика и перечислены основные угрозы. Целью работы были исследования преимуществ и недостатков средств сетевого анализа, а также выявления угроз и опасностей в их использовании.

Ключевые слова: сетевой анализ, трафик, анализатор, пакет, сниффер, сетевая безопасность, взлом, сетевой монитор.

Введение. Первоначально анализаторы сетевых пакетов, или снифферы, разрабатывались для решения определенных сетевых проблем. Их задачи сводились к перехвату, интерпретации и сохранению полученной информации для последующего анализа пакетов, передаваемых по сети. С одной стороны, это позволило системным администраторам и инженерам службы технической поддержки наблюдать за тем, какие данные передаются по сети, диагностировать и устранять возникающие проблемы и следить за общей безопасностью. В этом смысле пакетные снифферы представляют собой мощный инструмент диагностики сетевых проблем. С другой стороны, подобно многим другим мощным средствам, изначально предназначавшимся для администрирования, с течением времени снифферы стали использоваться для диаметрально противоположных задач, уже не столько для защиты сетевой инфраструктуры, а для атаки на нее. Действительно, сниффер в руках злоумышленника представляет собой довольно опасное средство и может использоваться для получения паролей, конфиденциальных данных и другой важной информации. Именно глубокое понимание возможностей сетевых анализаторов и позволяет проводить полнейшую диагностику вычислительной сети, а также иметь представление о ее уязвимостях.

Анализ средств сетевого анализа. В дальнейшем снифферы выросли в специализированные программные комплексы – сетевые мониторы. В некоторых источниках различают снифферы и мониторы на разные по классу программного обеспечения (ПО). В каждый монитор встроено как минимум несколько анализаторов протоколов. Но, несмотря на это, снифферы, зачастую, обладают более богатыми возможностями по пакетному анализу и предоставляют более детальную и объемную информацию, в то время как сетевые мониторы служат для автоматизации мониторинга и анализа за сетью. И, несмотря на некоторую схожесть, у них несколько разные задачи в системном администрировании.

Системы мониторинга используются для повседневного анализа и мониторинга сети, в то время как снифферы используются для точечного и разового анализа участка сети и анализа уже обнаруженной неисправности. Кроме того, мощные комплексы по мониторингу потребляют довольно много ресурсов и имеют высокую цену, в то время как снифферы зачастую представляю собой небольшие утилиты, использующиеся для более детального изучения конкретного трафика. С другой стороны граница между данным ПО довольно размыта, и очень часто используется обобщенный термин «сетевой анализатор».

В руках системного администратора подобное ПО является довольно полезным инструментом. Оно помогает находить и устранять узкие места, снижающие пропускную способность, обнаруживать проникновения, анализировать работу всей вычислительной сети. Однако стоит отметить, что для использования снифферов системный администратор должен обладать глубокими знаниями устройства сети и работы сетевых протоколов, иначе эффективность использования подобных средств для диагностики будет минимальной. При должном знании работы сетевых протоколов и структуры передаваемых пакетов можно диагностировать любую неисправность на любом сегменте сети [1].

С другой стороны процесс диагностирования неисправностей через снифферы достаточно трудоемкий. На анализ пакетов и протоколов может уйти не один день, поэтому снифферы редко используются для общего диагностирования неисправностей. Они применяются в первую очередь для диагностирования неисправности в работе того или иного протокола или для отслеживания перемещения трафика через тот или иной канал, однако ценность сетевых анализаторов переоценить сложно.

Для автоматизации анализа и мониторинга сетевого трафика применяются специальные комплексы и системы сетевого мониторинга. Данные средства зачастую включают в себя множество различных компонентов и подсистем, каждое из которых выполняет свою роль. Они используются для обнаружения вторжений, для мониторинга производительности сети или для обнаружения неисправностей и некорректной работы протоколов или оборудования. Очень часто их использование более удобно и требует меньше времени на диагностику и анализ, но такие комплексы очень часто стоят невероятно дорого, вследствие чего множество мелких и средних предприятий просто не может себе их позволить [2].

Сетевые мониторы позволяют вести общую статистику и учет по каждому протоколу и коллизиям, производить сложные эвристические анализы, имеют огромное количество средств оповещения, позволяют визуализировать мониторинг посредством графиков, диаграмм и т.д.

Опасности в использовании и угрозы от средств сетевого анализа. Сетевые анализаторы не всегда применяются в благих целях, иногда они могут представлять серьезную опасность. Специалисты в области компьютерной безопасности относят атаки на компьютеры при помощи анализаторов протоколов к так называемым атакам второго уровня. Это означает, что компьютерный взломщик уже сумел проникнуть сквозь защитные барьеры сети и теперь стремится развить свой успех. С помощью анализатора протоколов злоумышленник может попытаться перехватить различную конфиденциальную информацию, такую как: пароли, секретные финансовые данные, корпоративную почту, имена и пароли пользователей, и многое другое. Имея в своем распоряжении достаточные ресурсы, компьютерный взломщик в принципе может перехватывать любую информацию, передаваемую по сети. Помимо этого некоторые средства позволяют генерировать собственные пакеты данных, подделывать и перенаправлять трафик определенных протоколов или и вовсе представляться другим устройством сети [3].

Снифферы существуют практически для всех платформ и операционных систем. В то же время они, по сути своей, вообще независимы от родительской платформы, так как работают на более низком уровне и подвергаются анализу не конкретный компьютер, а

протоколы. В этой связи анализатор протоколов может быть установлен в любой сегмент сети и оттуда осуществлять перехват сетевого трафика, который в результате широковещательных передач попадает в каждый компьютер, подключенный к сети.

Но даже при использовании средств сетевого анализа в благих целях все равно существуют определенные риски и угрозы вычислительной сети. Сниферы, как и сетевые мониторы, таят в себе потенциальную опасность. Из-за того, что через них проходит огромное количество различной конфиденциальной информации, а их использование в системном администрировании происходит повсеместно – они стали объектом пристального внимания злоумышленников. С помощью эксплойтов, уязвимостей, бекдоров и деассемблирования данное ПО взламывается, а хакеры получают доступ к нужной им информации. Помимо этого, очень многий инструментарий позволяет хранить пакеты данных в некотором пуле, что также позволяет злоумышленникам получить контроль над огромным количеством информации. Механизмы пакетного анализа и мониторинга являются, по сути, лакомым кусочком для взломщиков любого рода, поскольку не просто дают возможность просмотреть любую информацию, но и незаметно красть целые объемы данных. Внешние атаки, использующие уязвимости в средствах мониторинга и анализа трафика, куда более опасны, так как, как правило, вычислить злоумышленника очень и очень сложно.

Хакеры и прочие злоумышленники не теряют времени даром. Они постоянно ищут все новые и новые способы получения интересующих их данных. Оказывается, способов много, начиная от вывода монитора из строя переполнением его буфера, заканчивая тем, что можно заставить монитор выполнить любую команду, посланную злоумышленником.

Существуют специальные лаборатории, анализирующие безопасность ПО. Их отчеты внушают тревогу, так как серьезные бреши находятся довольно часто и, что самое важное, в довольно крупных и известных приложениях и комплексах. Примером реальных брешей может послужить довольно известная программа RealSecure – коммерческая система обнаружения вторжения (IDS) от ISS.

Данное ПО вело себя нестабильно при обработке некоторых DHCP-сигнатур (DHCP_ACK – 7131, DHCP_Discover – 7132, и DHCP_REQUEST – 7133), поставляемых с системой. Посылая злонамеренный DHCP-трафик, уязвимость позволяет удаленному нападающему нарушить работу программы. Уязвимость обнаружена в Internet Security Systems RealSecure Network Sensor 5.0 XPU 3.4-6.5.

Способы защиты. Наибольшую опасность сетевые анализаторы представляли во времена, когда информация передавалась по сети в открытом виде, т.е. без шифрования, а локальные сети строились на базе концентраторов (хабов). Однако эти времена безвозвратно ушли, и в настоящее время использование снифферов для получения доступа к конфиденциальной информации – задача отнюдь не из простых.

Чем больше сегмент сети, тем больше информации можно получить с помощью снифферов и систем мониторинга. Именно поэтому в первую очередь необходимо разбивать сегменты сети на более мелкие подсети и уменьшать домены коллизий, препятствуя прослушиванию среды передачи данных. В современных реалиях это очень просто реализовать, поскольку устаревшие концентраторы давно заменены на коммутаторы, ограничивающие распространение широковещательных сигналов. Помимо этого, стоит отметить, что сегментация сети зачастую влечет к более удобному ее управлению и контролю, так как позволяет точно отследить источник возникновения неполадок, а сами неполадки не сказываются на большом участке сети.

Во вторых, можно использовать шифрование трафика, что усложняет считывание пакетных данных. Данный способ весьма хорош и эффективен, так как взломать сложные криптографические современные алгоритмы практически невозможно. Однако есть и недостатки: криптография ведет к увеличению трафика и к более сложной его обработке.

Постоянно шифровка и дешифровка сильно сказывается на общей производительности всей сети, поэтому очень часто от этих мер отказываются или применяют их лишь в некоторых сегментах. Кроме того, если анализатор был запущен до внедрения шифрования, он может перехватить первичные ключи обмена, или и вовсе выполнять роль скрытого посредника, обходя, таким образом, проблему шифрования [4].

В третьих, существуют контрмеры по анализу трафика – антиснифферы. Они работают по принципу измерения времени реагирования хостов на сетевые запросы и определении, не приходится ли хостам обрабатывать «лишний» трафик. Антиснифферы не могут полностью ликвидировать угрозу сниффинга, но они очень важны при построении комплексной системы защиты.

Для предотвращения использования средств самого системного администратора необходимо использовать пароли и учетные записи на системах мониторинга, тщательно следить за антивирусным ПО и ограничить физический доступ к вычислительному устройству, будь то персональный компьютер или специализированный сервер.

Заключение. На сегодняшний день средства анализа и мониторинга сети стали более надежными и безопасными, а контрмеры против их злоумышленного применения более гибкими и изощренными. Сегодняшние стандарты шифрования чрезвычайно затрудняют процесс перехвата конфиденциальной информации, а использование коммутаторов и антисниффинговые средства защиты добавляют проблем при использовании подобного программного обеспечения. В связи с этим в настоящее время пакетные снифферы постепенно утрачивают свою актуальность в качестве инструментов хакеров, но в то же время остаются действенным и мощным средством для диагностирования сетей. Более того, снифферы могут с успехом использоваться не только для диагностики и поиска сетевых неисправностей, но и для аудита сетевой безопасности. В частности, применение пакетных анализаторов помогает в обнаружении несанкционированного трафика, позволяет обнаружить и идентифицировать несанкционированное программное обеспечение, идентифицировать неиспользуемые протоколы для удаления их из сети, осуществлять генерацию трафика для испытания на вторжение (penetration test) с целью проверки системы защиты, работать с системами обнаружения вторжений (IDS) и многое другое.

С другой стороны ни одна система не является полностью безопасной и, несмотря на огромное количество контрмер, уязвимости и лазейки все же существуют и активно используются. Средства мониторинга уязвимы для взлома самого программного обеспечения с помощью вирусов и троянов, а шифрование и отслеживание несанкционированных снифферов не всегда дает положительный результат. Лишь комплексная защита и четкое следование протоколам безопасности [5].

Литература

1. Уилсон Э. Мониторинг и анализ сетей. Методы выявления неисправностей. – М.: ЛОРИ, 2012. – 386 с.
2. Олифер В., Олифер Н. Компьютерные сети. Принципы, технологии, протоколы. – 5-е изд. – СПб.: Изд. дом «Питер», 2017. – 992 с.
3. Основные принципы обнаружения сетевых атак [Электронный ресурс]. – Режим доступа: https://studref.com/339804/informatika/osnovnye_printsipy_obnaruzheniya_setevyh_atak.html (дата обращения: 19.01.2018).
4. Анализ сетевых пакетов [Электронный ресурс]. – Режим доступа: <http://compress.ru/article.aspx?id=16244> (дата обращения: 26.01.2018).
5. Павлов К.В. Сетевые анализаторы [Электронный ресурс]. – Режим доступа: <http://works.doklad.ru/view/P1miE5MOJqA.html> (дата обращения: 24.01.2018).

**Малярчук Людмила Константиновна**

Год рождения: 1980

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра аппаратно-программных комплексов
вычислительной техники, студент группы № S4100Направление подготовки: 09.04.01 – Информатика и вычислительная
техника

e-mail: g_mila@inbox.ru

**Авксентьева Елена Юрьевна**

Год рождения: 1981

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра аппаратно-программных комплексов
вычислительной техники, к.педагог.н., доцент

e-mail: avksentievaelena@rambler.ru

УДК 004.65

**РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ УЧЕТА, АНАЛИЗА
И ПЛАНИРОВАНИЯ ВАКЦИНАЦИИ НАСЕЛЕНИЯ****Малярчук Л.К.****Научный руководитель – к.педагог.н., доцент Авксентьева Е.Ю.**

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

Объектами исследования являлись автоматизированные информационные системы, применяемые в медицине. Предмет исследования – информационные системы для учета, анализа и планирования вакцинации населения.

Ключевые слова: информационная система, прививка, вакцинация, вакцинопрофилактика, иммунопрофилактика, календарь прививок, плановые прививки, журнал прививок, автоматизированная программа управления, оптимизация, автоматизированная система, МИС, ИС, АИС.

Введение. В настоящее время с развитием современных прогрессивных технологий, внедряемых повсеместно, применение медицинских автоматизированных информационных систем становится все более актуальным. Такие системы позволяют обеспечить контроль качества лечения, и оптимизируют деятельность лечебного учреждения.

В связи с огромным разнообразием прививок и постоянном его росте, без их учета уже никак не обойтись. На сегодняшний день существуют различные автоматизированные информационные системы с большим набором функций и средств, в полной мере позволяющие вести учет и планирование прививок.

Анализ существующих программных решений в городе Санкт-Петербурге показывает, что процесс учета прививок не полностью автоматизирован, имеет огромное количество недостатков. Наиболее эффективным решением является создание нового программного обеспечения, позволяющего повысить эффективность работы медицинских работников за счет создания более удобного и понятного графического интерфейса пользователя. Необходимо автоматизировать процесс учета прививок на всех его этапах, что позволит в рамках одной распределенной системы обеспечить решение всего круга задач по управлению иммунопрофилактикой, а следовательно, в повышении его эффективности на рабочем месте.

Весь управленческий процесс основывается на информационном обеспечении автоматизированных систем управления, которое, в свою очередь, позволяет учреждениям

более высокого уровня оценить эффективность деятельности всей системы здравоохранения и отдельной организации, а следовательно, своевременно принимать эффективные управленческие решения, направлять и мотивировать персонал для их исполнения, планировать ресурсную базу, реализовывать эффективные модели поведения, выбирать адекватный стиль руководства.

Эффективность организации процесса вакцинопрофилактики обуславливается наличием законодательной базы, оптимизацией системы управления, использованием современных информационных технологий и телекоммуникаций, обеспечением населения своевременной и достоверной информацией.

Автоматизированные системы управления иммунизацией позволяют улучшить количественные характеристики системы вакцинопрофилактики, решить задачи, направленные на ее качественное изменение, повысить полноту и своевременность сбора, анализа, хранения и передачи информации.

Из-за постоянной необходимости усовершенствования работы по вакцинопрофилактике выдвигаются все новые и новые требования ко всем медицинским учреждениям.

Внедрение в практику здравоохранения комплексной программы «Управление иммунизацией» в Санкт-Петербурге. Согласно приказу I20/I03a комитета по здравоохранению администрации Санкт-Петербурга и центра Госсанэпиднадзора в Санкт-Петербурге от 09.04.1998 [1] было произведено внедрение комплексной программы «Управление иммунизацией» в Санкт-Петербурге.

На персональные компьютеры пользователей была установлена программа «Вакцинопрофилактика». В организациях, где были установлены другие программы и в тех, которые еще не успели внедрить у себя программный продукт «Вакцинопрофилактика» был установлен программный продукт «Документ», обеспечивающий ручное заведение отчетных форм. Передача отчетов в Городской Центр Госсанэпиднадзора и в инфекционно-эпидемиологический организационно-методический отдел Комитета по здравоохранению Администрации Санкт-Петербурга из этих программ осуществлялась на магнитных дисках или по модемной связи.

На тот момент программа «Вакцинопрофилактика» научно-внедренческого товарищества «БИМК-Д» была не единственным программным продуктом, существующим в нашем городе. Из-за недоступности формата данных, передаваемых в Городской Центр Госсанэпиднадзора и в инфекционно-эпидемиологический организационно-методический отдел Комитета по здравоохранению Администрации Санкт-Петербурга, пользователям приходилось отказываться от остальных программных продуктов из-за неудобства ручного заведения отчетных форм, несмотря на преимущества самих программ учета прививок.

Также за счет средств Комитета по здравоохранению согласно источнику [2] была приобретена вычислительная техника непосредственно под внедрение программной системы «Управление иммунизацией», что тоже очень сильно повлияло на выбор программного обеспечения в пользу научно-внедренческого товарищества «БИМК-Д».

Еще одним фактом в пользу этого программного продукта является то, что только программный продукт научно-внедренческого товарищества «БИМК-Д» имеет сертификат Министерства здравоохранения Российской Федерации на его использование в учреждениях здравоохранения Российской Федерации.

На сегодняшний день ситуация в нашем городе особо не изменилась. Фирма-разработчик программы «Вакцинопрофилактика» до сих пор является лидером в этой области, так как отчеты, передаваемые в Городской Центр Госсанэпиднадзора и в инфекционно-эпидемиологический организационно-методический отдел Комитета по здравоохранению Администрации Санкт-Петербурга, принимаются только в том закрытом формате данных, который они когда-то разработали.

Недостатки существующей на сегодняшний день схемы учета прививок. Почти все программные продукты в этой области были разработаны для операционных систем (ОС) самых первых версий Windows и DOS. Под последние версии ОС Windows не адаптированы.

Формат передачи данных в Городской Центр Госсанэпиднадзора и в инфекционно-эпидемиологический организационно-методический отдел Комитета по здравоохранению Администрации Санкт-Петербурга не является общедоступным.

Невозможность работы системы в многопользовательском режиме.

Нет взаимосвязи комбинированных вакцин, т.е. оказывающих профилактическое воздействие сразу для нескольких инфекций соответствующим им инфекциям.

Также нет автоматической перенастройки планов профилактических прививок в ситуации изменения соответствующих календарей прививок.

Базы данных в каждом учреждении, осуществляющем учет прививок, изолированы друг от друга, нет единой базы данных, а следовательно, процесс учета прививок не полностью автоматизирован.

В детских дошкольных учреждениях, школах, и других учреждениях, осуществляющих вакцинопрофилактику, программы по учету прививок не установлены, что тоже доказывает не полную автоматизацию процесса.

У мобильных прививочных пунктов также не возможности вносить сделанные прививки в программу по их учету.

В существующих программных продуктах нет возможности брать сведения о пациентах из других баз. Это дублирует работу персонала по вводу общих данных о пациентах, таких как фамилия, имя, отчество, дата рождения и т.д. Также такие сведения, как медицинский отвод от прививки дает врач, и процесс передачи этих данных мог бы существенно упроститься, если бы была единая база пациентов, где врач мог сам в программу занести медицинский отвод.

Процесс учета вакцин происходит практически в ручном режиме. Уже давно самым распространенным и наиболее оптимальным методом предметно-количественного учета является штриховое кодирование.

В подавляющем числе программ картотек отсутствует произведенная экстренная иммунопрофилактика.

В этих программах не предусмотрено информирование пациентов о предстоящих прививках, что существенно влияет на привитость населения.

Процесс перевода пациента из одного лечебного учреждения в другое также не автоматизирован.

В базах таких программ нет информации о врачах, которые направляют на прививку, а также информации о медработниках, выполнивших эту прививку. Такой учет позволил бы свести к минимуму врачебные ошибки. Также появится возможность оценивать эффективность работы медицинского персонала. В спорных ситуациях можно будет быстро получить информацию о том, кто направил на прививку и кто ее выполнил.

Также в них нет справочника возможных осложнений после проведения вакцинации.

В организациях, запрашиваемых данные о прививках, нет возможности самим сделать запрос о прививках.

Почти во всех программах очень неудобный и непонятный интерфейс, что усложняет работу пользователя программного продукта.

Еще очень важным моментом для пользователя является понятная инструкция, которой в существующих программах либо нет, либо написана языком, не понятным обычному пользователю.

Существующие программы не имеют удаленной поддержки и сопровождения.

Согласно статье [3], набор основных функций современных систем автоматизации службы иммунопрофилактики ограничен возможностью регистрации произведенных прививок, а, в лучшем случае, возможностью автоматического планирования

профилактических прививок по календарям прививок (без возможности их ручной настройки).

Также авторы этой статьи считают, что комплекс программ «Управление иммунопрофилактикой» научно-внедренческого товарищества «БИМК-Д» (г. Санкт-Петербург) имеет ряд существенных недостатков. Во-первых, это отсутствие единого комплексного подхода к формированию регистра детского, подросткового и взрослого населения. Во-вторых, несколько затруднена процедура регистрации произведенных профилактических прививок. Данная процедура не позволяет медицинскому персоналу производить регистрацию реального объема произведенных профилактических прививок, с одновременной возможностью их дальнейшего планирования. Таким образом, эту задачу можно эксплуатировать в службе детской, подростковой или взрослой иммунопрофилактики, причем данные этих служб не находятся в едином информационном поле.

Заключение. К вышеперечисленным выводам о недостатках автор пришел путем анализа существующих программ, возможности которых удалось увидеть на практике, благодаря работе в двух таких программах, опроса пользователей, преимущественно тех, которым удалось работать в нескольких из этих программ и литературного обзора современных информационных технологий.

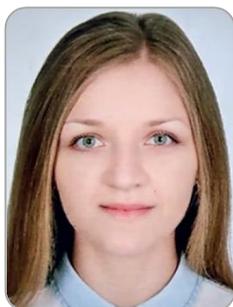
В результате исследования, он пришел к выводу, что в России необходимо разработать новую информационную систему для учета, анализа и планирования вакцинации населения. Для этого нужно создать единую базу данных для учета прививок и подключить к ней все медицинские организации, медпункты, мобильные прививочные пункты, а также все организации, которым необходима информация из этих баз. Это повысит эффективность хранения и обработки данных, любая организация, подключенная к такой базе, сможет получить необходимые данные сама и тем самым сократится количество документов, передаваемых на носителях информации. При этом надо осуществить связь новой базы с уже существующими базами в медучреждениях, для того чтобы была возможность обмениваться сведениями, такими как ФИО, дата рождения и т.д. Также необходимо разработать новый программный настраиваемый интерфейс для учета, анализа и планирования вакцинации населения для актуализации базы данных по запросу в автоматизированном режиме. Настройку интерфейса необходимо сделать, учитывая индивидуальные потребности каждого пользователя данной базой. Для процесса учета вакцин нужно задействовать штриховое кодирование. И включить такую функцию, как информирование пациентов о предстоящих прививках.

А также при разработке новой информационной системы необходимо избежать недостатков существующих баз.

Практическая значимость данной работы заключена в упрощении труда медицинских работников за счет создания более удобного и понятного графического интерфейса пользователя, автоматизации процесса учета прививок на всех его этапах, позволяющего в рамках одной распределенной системы обеспечить решение всего круга задач по управлению иммунопрофилактикой, а следовательно, в повышении его эффективности на рабочем месте.

Литература

1. Управление иммунизацией, Приказ Курчанова В.И. [Электронный ресурс]. – Режим доступа: <https://nvbimk.nethouse.ru/documents>, своб.
2. О распределении компьютеров по программе «Вакцинопрофилактика» [Электронный ресурс]. – Режим доступа: <http://docs.cntd.ru/document/9106320>, своб.
3. Опыт автоматизации деятельности службы иммунопрофилактики лечебно-профилактического учреждения (в рамках решения задачи комплексной автоматизации ЛПУ) [Электронный ресурс]. – Режим доступа: <http://www.mislpu.narod.ru/mislpu/art0919991024.htm>, своб.

**Новикова Евгения Игоревна**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра аппаратно-программных комплексов
вычислительной техники, студент группы № S4200Направление подготовки: 09.04.01 – Информатика и вычислительная
техника

e-mail: novenkaya_2010@mail.ru

УДК 004.416.6

**ИССЛЕДОВАНИЕ МЕТОДОВ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ
И МАСШТАБИРУЕМОСТИ ПРИКЛАДНЫХ РЕШЕНИЙ 1С****Новикова Е.И.****Научный руководитель – к.т.н., доцент Мартынов В.П.**

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

В работе рассмотрены: платформа «1С:Предприятие», которая поддерживает все прикладные решения и выполняет все задачи, связанные с функциональностью, производительностью, эргономикой и т.д.; методы повышения производительности и масштабируемости прикладных решений 1С. Даны некоторые рекомендации, которые будут пригодны в использовании организациями в самых различных областях применения.

Ключевые слова: 1С:Предприятие, производительность, масштабируемость, администрирование, сервер 1С:Предприятия.

«1С:Предприятие» – это универсальная облачная и локальная система программ для автоматизации финансовой и более широкой операционной деятельности компании. «1С:Предприятие» обладает широкими возможностями для удовлетворения разнообразных потребностей современного бизнеса. Это достигается благодаря «настраиваемости» – возможности настройки системы на основе конкретных потребностей компаний и их бизнес-процессов. Разница между платформой и прикладными решениями проста: платформа, как следует из названия, поддерживает все прикладные решения и выполняет все задачи, связанные с функциональностью, производительностью, эргономикой и т.п. Платформа, по существу, стоит одна, тогда как существует множество прикладных решений. Компания 1С и их партнеры постоянно внедряют новые прикладные решения [1].

Масштабируемость – это способность системы справляться с растущими требованиями и увеличением рабочей нагрузки. Соответственно масштабирование – это способ повышения производительности системы за счет ее масштабируемости.

Одним из основных критериев масштабируемости системы является способность поддерживать высокую производительность при увеличении количества решаемых задач, объема обрабатываемых данных и количества одновременно работающих пользователей в условиях пиковых нагрузок [2].

На производительность программ 1С оказывает влияние целый спектр факторов. Основные из них:

- способ работы с информационной базой (клиент-серверный или файловый);
- число одновременно работающих пользователей;
- объем информационной базы;
- пропускная способность локальной вычислительной сети;
- характеристики аппаратной части сервера и рабочего места пользователя [3].

Один из способов повысить производительность – увеличить мощность серверов. Другой способ – конфигурирование кластера серверов.

Масштабируемость кластера серверов – это возможность настройки кластера серверов в соответствии с изменением требований к его производительности. Другими словами, это способность добавления одной или нескольких систем к существующему кластеру, когда общая нагрузка кластера превышает его текущую производительность.

Серверный кластер обеспечивает отказоустойчивость для следующих сценариев: сбой сервера компьютера (включая сбой центрального сервера), прекращение рабочих процессов или временные потери физического соединения между рабочей станцией пользователя и кластером. Это возможно, поскольку имеется возможность использовать резервные кластеры и рабочие процессы резервного копирования. Некоторые внутренние операции кластера также направлены на повышение толерантности к потерям соединения.

Кластер выполняет автоматическую балансировку нагрузки между своими рабочими процессами на основе анализа их производительности в реальном времени. Системный администратор имеет возможность тонкой настройки балансировки нагрузки, ограничивая количество подключений или обработанных информационных баз в кластере или назначая часть функций кластера другим рабочим серверам.

В ходе изучения были выделены следующие способы масштабирования кластера серверов:

1. увеличить количество рабочих серверов, входящих в кластер – позволит использовать большее количество рабочих процессов, не увеличивая при этом нагрузку на каждый конкретный рабочий процесс;
2. использовать дополнительные менеджеры кластера и распределять сервисы между ними – позволит распределить всю нагрузку между несколькими процессами;
3. использовать дополнительные рабочие процессы на конкретном рабочем сервере – позволит снизить нагрузку на каждый конкретный рабочий процесс, более эффективно использовать аппаратные ресурсы рабочего сервера и позволит повысить надежность сервера, изолировав группы клиентов, работающих с разными информационными базами [4].

Требования к производительности процессора и к объему оперативной памяти, установленной на клиентском компьютере, в значительной мере зависят от характера задач, решаемых конкретным пользователем. Кроме того, следует учитывать, что процессор и оперативная память компьютера являются разделяемыми ресурсами, которые используются не только «1С:Предприятием», но и операционной системой, антивирусным программным обеспечением и другими программами, работающими на данном компьютере.

Требования к параметрам компьютера, на котором будет работать сервер «1С:Предприятия», будут зависеть от расположения сервера «1С:Предприятия» и сервера баз данных, например, MS SQL Server (на одном компьютере или на разных), от количества одновременно работающих пользователей, интенсивно выполняющих операции ввода информации или формирования отчетов, от объема обрабатываемой информации и от распределения нагрузки между сервером и клиентом.

В том случае, если сервер «1С:Предприятие» и MS SQL Server располагаются на одном компьютере, то требования каждого приложения к мощности вычислительных ресурсов компьютера суммируются. В этом случае основными факторами, влияющими на производительность, оказываются количество процессоров и их тактовая частота, а также объем оперативной памяти. Расположение серверов «1С:Предприятие» и MS SQL Server на разных компьютерах позволит администратору уменьшить нагрузку на систему компьютера путем ее распределения между серверами. Например, при выполнении сложных запросов, программа будет получать только необходимую ей выборку, а вся промежуточная обработка будет выполняться на сервере. Это позволит снизить требования к клиентским компьютерам и, соответственно, их стоимость, но при этом возрастут требования, предъявляемые к компьютеру, на котором установлен сервер «1С:Предприятия».

Основными факторами, влияющими на производительность MS SQL Server, являются объем информационной базы, количество одновременно работающих пользователей и вид задач, решаемых пользователями (создание документов, построение отчетов и т.п.). Следовательно, для оценки характеристик сервера (тактовой частоты, количества процессоров и объема оперативной памяти), необходимых для обеспечения требуемой пропускной способности и времени реакции системы, следует учитывать выше перечисленные факторы.

SQL Server использует оперативную память в основном для кэширования операций чтения/записи при выполнении действий с базой данных, поэтому увеличение объема оперативной памяти – неотъемлемая задача, связанная с ростом объема обрабатываемой информации. Иначе существенного снижения скорости выполнения операций не избежать, что приведет к значительному снижению производительности системы в целом.

Выбирая дисковую подсистему, отдавайте предпочтение дискам, объединенным в RAID-массив. По сравнению с организацией дискового массива средствами операционной системы, использование RAID-контроллера для этих целей является предпочтительным вариантом, поскольку позволяет построить более надежное и производительное решение [5].

Практическая значимость исследования состоит в том, что результаты данного исследования могут быть пригодны в использовании организациями в самых различных областях применения.

Литература

1. Об 1С:Предприятие 8 [Электронный ресурс]. – Режим доступа: <http://1c-germany.com/ru/about-1c-enterprise/>, своб.
2. Масштабируемость [Электронный ресурс]. – Режим доступа: http://v8.1c.ru/overview/Term_000000583.htm/, своб.
3. Производительность и масштабируемость 1С:Предприятия 8.1 [Электронный ресурс]. – Режим доступа: http://center-comptech.ru/st_proizvoditelnost_mashtabir.html/, своб.
4. Кластер серверов [Электронный ресурс]. – Режим доступа: <https://its.1c.ru/db/metod8dev#content:1525:hdoc/>, своб.
5. Рекомендации по выбору конфигурации компьютера для работы с «1С Предприятием 8» [Электронный ресурс]. – Режим доступа: <http://garant1c.ru/recom.html/>, своб.



Петров Валерий Владимирович

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра аппаратно-программных комплексов
вычислительной техники, студент группы № S4100

Направление подготовки: 09.04.01 – Информатика и вычислительная
техника

e-mail: valery.petrov@nordigy.ru



Костеж Владимир Александрович

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра аппаратно-программных комплексов
вычислительной техники, к.т.н., доцент

e-mail: k@limtu.ru

УДК 004.75

**ИСПОЛЬЗОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ TICK, ELK, ZABBIX
ДЛЯ МОНИТОРИНГА ИНФОРМАЦИОННЫХ СИСТЕМ**

Петров В.В.

Научный руководитель – к.т.н., доцент Костеж В.А.

В работе рассмотрено общее понятие мониторинга информационных систем в качестве сервиса, также выделены основные цели и задачи. В том числе были выделены ключевые особенности самых инновационных и популярных систем мониторинга с их последующим анализом.

Ключевые слова: система мониторинга, Zabbix, TICK, ELK, Telegraf, Influxdb, Chronograph, Capacitor, Elasticsearch, Logstash, Kibana.

Введение. В общем понятии мониторинг означает непрерывный процесс наблюдения и регистрации параметров объекта, с целью их удержания в заданных пределах в соответствии с заданными критериями. В общем случае ставится задача мониторинга компонентов информационной системы (ИС). В качестве параметров объекта в этом случае выступают загруженность ЦП, сетевых интерфейсов, дисков и т.п.

Цель мониторинга – обеспечение бесперебойной работы ИС предприятия за счет своевременного обнаружения неисправностей, сбоев и отказов оборудования и сетевых компонентов системы.

Общими задачами систем мониторинга служат:

- слежение за метриками ИС;
- запись, хранение, и обработка данных мониторинга, полученных от узлов ИС;
- оптимизация и облегчение работы IT-отдела;
- поиск проблемных мест в ИС.

Обеспечение контроля за ИС. Система мониторинга должна обеспечивать контроль за основными узлами в сети. Задача слежения за метриками ИС, состоит в том, что у определенных узлов сети есть свои, заданные метрики. У всех узлов сети есть базовые метрики и уникальные. Базовые метрики – это параметры, за которыми необходимо наблюдать у всех серверов, независимо от задач, которые они выполняют. Уникальные метрики зависят от задач, которые выполняет непосредственно сам узел сети. Очевидно, что мониторинг базы данных (БД) не требуется серверу, не обслуживающему БД. К базовым

метрикам относятся: загруженность центрального процессора (ЦП), оперативной памяти, сетевого интерфейса и т.п. К уникальным метрикам относятся: работоспособность RAID массива, контроль за БД, и т.п. Запись, хранение, и обработку данных, полученных от узлов ИС, система мониторинга записывает, хранит и обрабатывает. Все это хранится в БД. И на основе этих данных можно использовать как встроенные средства для анализа данных самой системы, так и стороннее программное обеспечение (ПО).

Методы исследования. Главный метод исследования – это анализ работы данных ИС на Lab-environment. Что включает в себя следующее:

1. рассмотрение и изучения технической документации;
2. анализ сильных и слабых сторон данных ИС;
3. тестирование данных ИС в условиях Lab-environment.

Результаты. В результате проведенного исследования можно сделать вывод – для системного мониторинга необходимо использовать несколько систем мониторинга (MaaS, Monitoring as a Service), которые будут выполнять свой, строго определенный набор задач. Это связано с тем, что не существует одной универсальной системы мониторинга, которая выполняла бы все функции. Есть лишь отдельные системы, который выполняют ограниченный спектр задач.

Обсуждение. Самыми инновационными и ведущими в области системного мониторинга являются следующие программные пакеты: ZABBIX, TICK, ELK.

Пакет ZABBIX – свободная система мониторинга и отслеживания статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования.

ZABBIX поддерживает несколько видов мониторинга:

- Simple checks – может проверять доступность и реакцию стандартных сервисов, таких как SMTP или HTTP, без установки какого-либо ПО на наблюдаемом хосте [1];
- ZABBIX agent – может быть установлен на UNIX-подобных или Windows-хостах для получения данных о нагрузке процессора, использования сети, дисковом пространстве и т.д.

External check – выполнение внешних программ. ZABBIX также поддерживает мониторинг через SNMP.

Пакет TICK – это набор ПО для мониторинга компонент ИС. Он состоит из следующего ПО:

- Telegraf;
- InfluxDB;
- Chronograf;
- Karactior.

Telegraf – это агент для сбора, обработки и агрегирования собираемых метрик с хоста для дальнейшей пересылки в Influxdb. Он написан на языке Go.

Telegraf также поддерживает плагины, и имеет концепцию 4-х видов плагинов:

1. Input plugins (плагины ввода) – собирают метрики из системы, служб, а также сторонних API;
2. Processors plugins (плагины процессинга) – трансформируют, преобразуют к одному виду, или фильтруют метрики;
3. Aggregator plugins (плагины агрегирования) – создают агрегированные показатели, например, среднее, минимальное, максимальное и т.д.;
4. Output plugins (плагины вывода) – записывают метрики в разные ИС [2].

InfluxDB – БД для хранения временных рядов, метрик и информации о событиях.

В числе преимуществ InfluxDB в первую очередь нужно выделить следующие:

- отсутствие зависимостей (следствие того, что она написана на Go);

- возможность работы, в том числе и в кластерном режиме;
- наличие библиотек для большого числа языков программирования (Python, JavaScript, PHP, Haskell и других);
- SQL-подобный язык запросов, с помощью которого можно производить различные операции с временными рядами (объединение, слияние, разбиение на части);
- удобный графический интерфейс для работы с БД.

Записывать данные в InfluxDB можно различными способами. Во-первых, данные в формате JSON можно передавать через HTTP API. Во-вторых, InfluxDB поддерживает протокол Carbon, используемый в инструменте для обработки и визуализации данных Graphite. В-третьих, данные можно отправлять по протоколу UDP [2].

Шардирование как в обычном, так и в кластерном режиме.

Chronograf – веб-приложение для управления БД InfluxDB. Его можно использовать для визуализации информации о данных системы мониторинга, создания алертинга или правил автоматизации.

Основные особенности:

- мониторинг инфраструктуры;
- управление алертами;
- визуализация данных;
- управление базой данных.

Kapacitor – платформа обработки данных с открытым исходным кодом, которая позволяет легко создавать оповещения, запускать задания ETL (Extract, Transform, Load), и обнаруживать аномалии.

Ключевые особенности:

- обработка как потоковых, так и пакетных данных;
- можно запрашивать данные из InfluxDB по расписанию, или получать данные через line протокол;
- возможность выполнять любые преобразования с данными, которые доступны в InfluxQL;
- хранение данных в InfluxDB;
- добавление пользовательских функций для обнаружения аномалий;
- встроенная интеграция с HipChat, OpsGenie, Alerta, Sensu, PagerDuty, Slack и т.д. [2].

ELK – это набор ПО для сбора, агрегирования, и удобного представления лог-файлов, метрик, событий и т.д. Огромная часть работы службы эксплуатации связана с мониторингом текущей инфраструктуры. Разработчики и системные администраторы могут оперативно пользоваться информацией, полученной из лог-файлов, которые были созданы приложениями, операционной системой и т.д.

Elasticsearch – высокомасштабируемый полнотекстовый поисковый и аналитический движок с открытым исходным кодом (основанный на Lucene). Он позволяет хранить, искать и анализировать большие объемы данных в реальном времени.

Ключевые особенности:

- кластеризация (возможность горизонтально увеличивать количество нод);
- полнотекстовый поиск в реальном времени (благодаря поисковому движку Lucene);
- шардирование и репликация данных (вероятность потери данных очень низка);
- экономное хранение документов на диске благодаря сжатию;
- открытый API с возможностью управлением всем сервисом;
- интеграция с Hadoop из коробки [3].

Logstash – ПО для обработки данных с открытым исходным кодом на стороне сервера, он может динамически унифицировать данные из разрозненных источников и нормализовать данные в выбранных вами направлениях.

- обработка лог-файлов на лету (например, веб-сервера Apache), и журналов приложений (таких как log4j2 для Java) с фильтрацией только нужных вам данных;

- сбор метрик из Ganglia, collectd, NetFlow, JMX, и многих других прикладных и инфраструктурных платформ через TCP и UDP;
- это стандарт сети сбора событий для приема данных, отправляемых с мобильных устройств, в интеллектуальные дома, транспортные средства, датчики здравоохранения и многие другие отраслевые приложения [3].

Kibana – предоставляет удобный GUI для использования elasticsearch, а также позволяет визуализировать все данные, которые хранятся в elasticsearch.

Основные особенности:

- возможность создавать графики и дашборды на основе данных в elasticsearch;
- возможность создавать алерты на основе дашбордов;
- обнаружение аномалий с помощью машинного обучения;
- открытый API;
- возможность управлять потоками данных logstash [3].

Как показывает исследование, следующие системы мониторинга имеют очень схожий функционал: TICK и ZABBIX. Но при детальном рассмотрении – данные системы очень разнятся в последующей их эксплуатации. Когда количество объектов будет расти, для поддержки ZABBIX нужно будет гораздо больше ресурсов, нежели для поддержки TICK. Но и у TICK есть существенный минус – так как это новый стек мониторинга, то количество шаблонов мониторинга, существующих плагинов и т.п., намного меньше, чем у ZABBIX [4].

Заключение. При работе с данными ИС, автор выделил для себя следующую главную особенность – на данный момент не существует готового программного продукта, который покрывал бы весь спектр задач современной ИТ-компания. Необходимо использовать несколько программных продуктов, для того чтобы решить все задачи.

Благодарности. Автор выражает свою благодарность компании RingCentral.

В частности: Головину В.А., а также Смирнову А.Б. за возможность подробно изучить и протестировать данные программные продукты.

Литература

1. Vacche A.D. Mastering Zabbix. – Second Ed. – Publisher: Packt Publishing Release, 2015. – 412 p.
2. Influxdb [Электронный ресурс]. – Режим доступа: <http://influxdata.com>, своб.
3. Elasticsearch [Электронный ресурс]. – Режим доступа: <http://elastic.co>, своб.
4. Uytterhoeven P. Zabbix Cookbook. – Packt Publishing Release, 2015. – 247 p.



Платунова Светлана Михайловна

Год рождения: 1960

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра аппаратно-программных комплексов
вычислительной техники

e-mail: psm@limtu.ru

УДК 004.72

**ТЕХНОЛОГИИ ОБЪЕДИНЕНИЯ ПАРАЛЛЕЛЬНЫХ КАНАЛОВ ПЕРЕДАЧИ
В СЕТЯХ ETHERNET**

Платунова С.М.

В работе освещены технологии, используемые при объединении параллельных каналов передачи в проводных и беспроводных сетях Ethernet.

Ключевые слова: сети передачи данных, технологии передачи данных, параллельные каналы передачи.

Объединение параллельных каналов передачи в сетях передачи данных [1] используются давно и успешно, так как имеют неоспоримые преимущества для повышения отказоустойчивости [2] и производительности как основных характеристик передачи данных.

Целью работы являлось формирование теоретических знаний по объединению параллельных каналов передачи данных в проводных сетях Ethernet.

Для выполнения данной работы были поставлены следующие задачи:

- познакомиться с основными теоретическими понятиями объединения параллельных каналов передачи данных;
- изучить принципы объединения параллельных каналов и линий связи в сетях Ethernet.

Агрегирование каналов в сетях передачи данных – это объединение нескольких физических каналов в один логический канал.

Повышение производительности происходит следующим образом. В идеале полоса пропускания канала становится равной сумме полос пропускания каналов, входящих в агрегированный канал. Такой агрегированный канал у различных производителей сетевого оборудования называется иначе транком, связанным каналом или склеенным каналом.

Повышение надежности происходит следующим образом: при отказе одного или нескольких каналов транка трафик без прерывания передачи будет передаваться через оставшиеся работоспособные каналы передачи. Это свойство также называется горячим резервированием.

Известными технологиями объединения каналов передачи являются агрегирование каналов и линий связи. Агрегирование восьми каналов часто оказываются более дешевым вариантом объединения, чем на порядок более производительный один канал. Все порты агрегированного канала должны быть одного типа – витая пара, одномодовое или многомодовое волокно, одинаковая скорость передачи, режим дуплекса и настройки управления потоком.

Агрегируют каналы обычно в сетевых коммутаторах, но можно производить объединение сетевых карт, например, в операционной системе Linux.

Решения объединенных каналов определяется в стандарте IEEE 802.3ad, но существуют и фирменные решения. Стандарт IEEE 802.3ad в составе группы стандартов для локальных вычислительных сетей IEEE 802 принят в 2000 году, полное название – «802.3ad Link aggregation for parallel links».

Агрегирование каналов на коммутаторе без дополнительных ухищрений приводит к переполнению таблицы коммутации и отказу сети передачи. Для решения этой задачи были введены протоколы управления агрегированием LACP (Link Aggregation Control Protocol). Протокол LACP через заданные таймауты проверяет параллельные каналы на работоспособность, и при отказе какого-либо канала, принадлежащего транку, помечает отказавший канал и перестает передавать по отказавшему каналу данные. Таким образом, сохраняется канал передачи.

При настройке на коммутаторе протокола LACP необходимо указать, на основании каких параметров коммутатор будет распределять трафик между каналами/портами транка. Для этого имеются 6 критериев:

1. Src-mac – базируется на основе MAC-источника;
2. Dst-mac – базируется на основе MAC-назначения;
3. Src-dst-mac (значение по умолчанию) – базируется одновременно на основе MAC-источника и MAC-назначения;
4. Src-ip – базируется на основе IP-источника;
5. Dst-ip – базируется на основе IP-назначения;
6. Src-dst-ip – базируется одновременно на основе IP-источника и IP-назначения.

Если входящие пакеты не имеют информации об IP при выборе критерия, например, Src-ip, тогда разделение нагрузки между портами в одной группе агрегирования будет основываться на критерии Src-mac. Аналогично при выборе других критериев, основанных на базе IP:

- src-ip -> src-mac
- dst-ip -> dst-mac
- src-dst-ip -> src-dst-mac

Таким образом, в зависимости от оборудования и трафика весь трафик может пойти по одному каналу. Это называется разделение нагрузки Load Sharing. Load Sharing равномерно распределяет таблицу MAC-адресов между портами, входящими в агрегированное соединение. Кадры с одинаковыми MAC-адресами назначения будут передаваться по одному физическому каналу агрегированного соединения без увеличения скорости передачи. Увеличение скорости передачи технология Load Sharing дает при передаче по агрегированному соединению кадром с несколькими MAC-адресами назначения.

В работе были рассмотрены технологии объединения параллельных каналов и линий связи, таких как протоколы управления агрегированием и разделения нагрузки. Это перспективное направление построения сетей передачи данных успешно развивается и применяется в сфере технологий объединения параллельных каналов и в проводных и беспроводных сетях.

Литература

1. Платунова С.М. Модель корпоративной сети при агрегировании каналов и резервировании линий // Научно-технический и производственный журнал «Вестник компьютерных и информационных технологий». – 2011. – № 2. – С. 55–59.
2. Демина А.С., Шалобаев Е.В. Аспекты надежности мехатронной системы защиты быстровращающегося оборудования // Сборник трудов V Всероссийского конгресса молодых ученых материалы конгресса: в 2 томах. – 2016. – С. 134–137.



Тазетдинов Аркадий Андреевич

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра аппаратно-программных комплексов
вычислительной техники, студент группы № S4100

Направление подготовки: 09.04.01 – Информатика и вычислительная
техника

e-mail: arkady738@bk.ru



Костеж Владимир Александрович

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра аппаратно-программных комплексов
вычислительной техники, к.т.н., доцент

e-mail: k@limtu.ru

УДК 004.654

**ОБОСНОВАНИЕ ВЫБОРА БАЗЫ ДАННЫХ ДЛЯ ДИНАМИЧЕСКИ
ИЗМЕНЯЮЩИХСЯ ОБЪЕМОМ ИНФОРМАЦИИ**

Тазетдинов А.А.

Научный руководитель – к.т.н., доцент Костеж В.А.

В работе был рассмотрен выбор между двух основных направлений технологий баз данных, SQL и NoSQL. Были рассмотрены основные положительные стороны, и проведены сравнения по ключевым параметрам. Были определены условия выбора одной из технологий.

Ключевые слова: SQL и NoSQL, масштабируемость, индексация, CRM-системы.

Сегодня огромное количество приложений используют базы данных (БД), будь то сервера, мобильные устройства или компьютеры. БД нужны как для игр, так и для бизнес-проектов. Все они обрабатывают определенный набор данных, заложенный на этапе разработки. Система управления БД – это программное обеспечение, предназначенное для хранения, управления и удаления данных. Для решения подобных задач разрабатывалось все больше и больше систем управления базами данных (СУБД) и программ для работы с ними [1–4].

SQL или NoSQL. Стоит отметить, что выражение «NoSQL» имеет стихийное происхождение и не имеет общепризнанного определения. Это название показывает вектор развития информационных технологий в противоположную сторону от реляционных баз данных. Расшифровывается данное выражение как Not Only SQL, хотя существуют сторонники и прямого определения No SQL. На данный момент существует два основных направления технологий баз данных. Реляционные и нереляционные базы данных. Или их можно назвать SQL (Structured Query Language) и NoSQL.

Различие в том, как они спроектированы, как хранят информацию и какие типы поддерживают. Реляционные базы хранят структурированные данные, которые обычно представляют объекты, сгруппированные в таблицах, формат которых задан на этапе проектирования. Нереляционные базы позволяют иметь сущность с произвольным набором атрибутов. В реляционной базе данных будет разбито на несколько взаимосвязанных таблиц, в нереляционной может храниться в виде единой сущности. В NoSQL-базах, в отличие от реляционных, структура данных не регламентирована – в отдельной строке или документе

можно добавить произвольное поле без предварительного декларативного изменения структуры всей таблицы. Таким образом, если появляется необходимость поменять модель данных, то единственное достаточное действие – отразить изменение в коде приложения.

Традиционные СУБД с SQL ориентируются на требования ACID к транзакционной системе: атомарность, согласованность, изолированность, надежность, тогда как в NoSQL вместо ACID может рассматриваться набор свойств BASE:

- гибкое состояние – состояние системы может изменяться со временем, даже без ввода новых данных, для достижения согласования данных;
- базовая доступность – каждый запрос гарантированно завершается (успешно или безуспешно);
- согласованность в конце – данные могут быть не согласованы, но приходят к согласованию в конечном итоге.

О выборе SQL-баз данных. NoSQL-базы стали популярными благодаря быстродействию и хорошей масштабируемости, но в некоторых ситуациях наилучшим решением может оказаться структурированные SQL-хранилища. Вот несколько причин, которые могут указывать, что лучшим решением выбрать SQL-базы:

1. необходимость соблюдения требований ACID (Atomicity, Consistency, Isolation, Durability – атомарность, непротиворечивость, изолированность, долговечность). Это позволяет уменьшить вероятность непредсказуемого поведения системы, а также обеспечить целостность. Достигается путем определения, как именно транзакции будут взаимодействовать с БД;
2. данные, с которыми вы будете работать, структурированы, при этом структура не будет подвержена частым изменениям;
3. приложение не сильно привязывается к конкретной СУБД. Язык SQL универсален для реляционных хранилищ, и пользователю не придется кардинально переписывать код в случае смены СУБД;
4. SQL имеет богатую историю и множество стандартов. Это сильный и сложный инструмент, который позволяет проводить сложные операции с данными.

О выборе NoSQL-баз данных. Если есть основания полагать, что БД будет работать с большими объемами информации, стоит посмотреть в сторону NoSQL-баз. Основными преимуществами у них являются:

- хранение больших объемов неструктурированной информации. База данных не накладывает ограничений на типы хранимых данных. Также при изменении условий задачи в процессе работы можно добавить новые типы данных;
- использование хранилищ и облачных вычислений. Но они требуют, чтобы данные можно было легко распределить между несколькими серверами для обеспечения масштабирования;
- быстрая разработка. Если вы разрабатываете систему, используя agile-методы, применение реляционной БД может замедлить работу. NoSQL-базы данных не нуждаются в том же объеме подготовительных действий, которые обычно нужны для реляционных баз;
- простота работы. Многие NoSQL-решения в основном хранилища вида «ключ-значение» имеют по сравнению с реляционными БД очень сильно урезанную функциональность. В таком случае оператору базы данных не требуется глубоких знаний достаточно мощного и гибкого механизма работы с SQL-запросами. Это снижает входной порог для начала работы с NoSQL-хранилищами;
- более простой синтаксис запросов, следовательно, меньше ошибок. Для упрощения работы с БД некоторыми разработчиками используется ORM-технология, позволяющая автоматически транслировать операции с объектами в запросы к базе данных. Язык SQL универсален и очень емок, для полноценной работы с ним необходим определенный багаж

знаний, но языки запросов современных NoSQL-хранилищ больше подходят для выполнения простых манипуляций с БД.

Из-за данных преимуществ NoSQL-баз данных, MongoDB, RavenDB, CouchDB, Redis стали популярны.

Масштабируемость. Под масштабируемостью, некоторые могут подразумевать репликацию, так что когда мы говорим о масштабируемости – мы имеем в виду автоматическое распределение данных между несколькими серверами, если вы используете объем данных, который не может быть обработан на одной машине, или если вы не хотите управлять распределением вручную. Одно из основных различий между этими технологиями в том, что NoSQL-базы проще и лучше масштабируются. Масштабируются и SQL-базы, но это потребует гораздо больше человеческих затрат и аппаратных ресурсов.

- Хранилище типа «ключ-значение». Используется для приложений с одним типом объектов в ситуациях, когда поиск объектов выполняют лишь по одному атрибуту.
- Хранилище, ориентированное на документы. Подходит для хранения объектов различных типов. Подходит для приложений, в ходе работы с которыми допускается реализация принципа «согласованность в конце» с ограниченными атомарностью и изоляцией. Рекомендуется применять механизм кворумного чтения для обеспечения своевременной атомарной.
- Система хранения данных с расширяемыми записями. Более высокая пропускная способность и лучшие возможности параллельной обработки данных ценой слегка более высокой сложности, нежели у хранилищ, ориентированных на документы. Для упрощения разделения данных используются HBase или Hypertable.
- Масштабируемая RDBMS. Использование семантики ACID освобождает программистов от необходимости работать на достаточно низком уровне, а именно, отвечать за блокировки и непротиворечивость данных, обрабатывать устаревшие данные, коллизии. Стоит обратить внимание на такие системы, как MySQL Cluster, VoltDB, Clustrix, ориентированные на улучшенное.

Индексация. В системах RDBMS индексация используется для ускорения операций извлечения данных из баз. Отсутствие индекса означает, что таблица должна быть просмотрена целиком, чтобы выполнить запрос на чтение. И в SQL, и в NoSQL-базах индексы служат для ускорения, оптимизирования и извлечения данных. Но то, как именно они работают – различается из-за разных архитектур баз данных и особенностей хранения информации в базе. В то время как SQL-индексы представлены в виде B-деревьев, которые отражают иерархическую структуру реляционных данных, в NoSQL-базах данных они указывают на документы, или на части документов, между которыми в основном нет никаких отношений.

CRM-приложения – это один из лучших примеров систем, для которых характерны огромные объемы ежедневно обрабатываемых данных и очень большое количество транзакций. Большая часть данных транзакций все еще хранится в SQL-базах, применение находят общедоступные системы класса DBaaS, в результате серьезная нагрузка по обработке данных может быть перенесена в облачные NoSQL-базы. Использование таких служб позволяет разработчику не решать задачи по обслуживанию хранилищ, это, кроме того, область, где NoSQL-базы используются для глубоко анализа данных. Основное преимущество подобных систем – использование неструктурированных данных, похожих на стандартные документы. Такие данные подаются на вход статистических моделей, которые позволяют выполнить разные виды анализа.

Заключение. Занимаясь поиском системы управления БД, можно выбрать одну технологию, а позже, уточнив требования, переключиться на что-то другое. Вот признаки

проектов, для которых идеально подойдут SQL-базы:

- имеются логические требования к данным, которые могут быть определены заранее;
- важна целостность данных;
- необходим стандарт, который хорошо себя зарекомендовал;
- большой опыт работы и богатая техническая поддержка;
- возможная необходимость быстрого переноса с одной СУБД на другую.

А вот свойства проектов, для которых подойдет что-то из сферы NoSQL:

- требования к начальным данным неопределенны;
- данные проекта будут вероятнее всего корректироваться со временем, но начинать разрабатывать необходимо сейчас;
- необходимость иметь огромную скорость обработки данных и хорошую масштабируемость.

По мнению автора, при выборе SQL или NoSQL надо отталкиваться от поставленной задачи. Если они могут быть логически связаны, то лучше использовать SQL. При всех плюсах NoSQL нужно понимать, что для работы с ней нужно больше опыта, необходимо будет продумывать множество нюансов для корректной работы. С NoSQL связаны высокие риски. Проблемы с драйверами, сырость кода, иногда недостающие инструменты (различные прокси, инструменты резервного копирования и т.д.). Использование Java, Go и прочих высокоуровневых языков, на которых любят писать NoSQL, к сожалению, приводит к непредсказуемому времени ответа СУБД, и обходится эта проблема весьма нетрадиционно. NoSQL-решения весьма прожорливы в плане ресурсов. И не так много людей смогут вам подсказать, почему что-то не работает в той или иной ситуации.

Литература

1. Lisovenko A. NoSQL базы данных: понимаем суть [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/152477/>, своб.
2. Cattell R. Scalable SQL and NoSQL Data Stores [Электронный ресурс]. – Режим доступа: <http://www.cattell.net/datastores/Datastores.pdf>, своб.
3. Ghosh M., Gupta I., Gupta S. Fast Compaction Algorithms for NoSQL Databases [Электронный ресурс]. – Режим доступа: <http://web.engr.illinois.edu/~sgupta49/papers/compaction.pdf>, своб.
4. NoSQL Database Explained [Электронный ресурс]. – Режим доступа: <https://www.mongodb.com/nosql-explained>, своб.



Трубицин Александр Михайлович

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра аппаратно-программных комплексов
вычислительной техники, студент группы № S4200

Направление подготовки: 09.04.01 – Информатика и вычислительная
техника

e-mail: leha.uuu@yandex.ru

УДК 004.056.5

**ИССЛЕДОВАНИЕ ПРОЦЕССА БЕЗОПАСНОГО ИСПОЛЬЗОВАНИЯ СЕТЕВЫХ
ПРОТОКОЛОВ**

Трубицин А.М.

Научный руководитель – к.т.н. Костеж В.А.

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

В работе рассмотрено обнаружение уязвимостей системы на каждом ее подуровне, а также проанализирована возможность применения методики экспертного анализа для оценки возможного ущерба и выработки конечных рекомендаций.

Ключевые слова: информационная безопасность, информационная угроза, экспертный анализ, экспертный подход, экспертная оценка.

Актуальность выбранной темы обосновывается необходимостью повышения уровня сетевой безопасности в рамках рационального использования ресурсов. Коммуникация между вычислительными станциями, расположенными на определенном расстоянии, осуществляется посредством конкретных сетевых протоколов, позволяющих получить доступ к ресурсам удаленной машины, а также инструменты управления информацией, обеспечивая такие параметры как: доступность, целостность, конфиденциальность [1–5].

Целью данной работы являлось выявление закономерности между уровнем выделяемых ресурсов и уровнем безопасности использования сетевых протоколов с выработкой рекомендаций.

С точки зрения степени вовлеченности предприятия и конкурентов в информационный процесс, можно рассматривать внутренние и внешние факторы, приводящие к потере защищаемой информации.

Внутренние факторы или действия по различным причинам или условиям создаются персоналом, внешние – конкурентами, направленные на получение коммерческой тайны. К таким факторам относятся:

- распространение засекреченной информации ее обладателем;
- утечка конфиденциальных данных по техническим каналам;
- неправомерный доступ к засекреченной информации различными методами.

Целесообразно идентифицировать не только обнаружившиеся угрозы, но и их происхождение, обеспечив тем самым выбор вспомогательных инструментов. После определения угрозы необходимо оценить вероятность ее реализации. Анализируя вероятность реализации угроз, разумно полагаться не только на среднестатистические значения, но и на специфику данной информационной системы.

На этой стадии выгодно использовать разработанную инфологическую схему объекта. Для малого предприятия рассматривается вся информационная схема с позиции обнаружения слабых узлов системы. Однако если компания крупная, то такой мониторинг может занять достаточно длительное время. В такой ситуации необходимо обратить особое

внимание на основные сервисы, предварительно согласившись с предположительной окончательной оценкой.

Уязвимым является любой элемент интегрируемой системы (ИС). Но зона мониторинга не может охватить каждый байт, поэтому возникает необходимость анализировать пошагово, на каждом уровне детализации, при этом используя приближенную оценку.

Исследования показали существование конкретизированных зависимостей между затратами ресурсов и полученным уровнем безопасного использования сетевых протоколов. К промежуточным результатам можно отнести исследования ресурсоемкости использования протоколов SSH и SFTP. К практическим результатам относится конкретизация определения уровня трудозатрат при подготовке к использованию и применению указанных протоколов для повышения уровня безопасности вычислительной сети.

При проведении исследований была рассмотрена уязвимость элементов ИС состоящей из 4-х уровней:

1. внешний уровень характеризует взаимодействие между подсистемами ИС. На этом уровне подсистема предоставляет информационные сервисы, а также доступ к ним по требованию других подсистем. Все попытки незаконного вторжения в подсистему как со стороны внешнего воздействия, так и персонала, имеющего доступ к сегментам сети, должны пресекаться на данном уровне;
2. сетевой уровень характеризует доступ к информационным потокам данных между локальными сетями подсистемы. Авторизация, аутентификация, идентификация – все эти инструменты проверки подлинности обеспечивают надежную защиту данных на данном этапе;
3. системный уровень регулирует доступ к средствам операционной системы. На этом уровне осуществляется напрямую управление оборудованием предприятия, корректируется или устанавливается конфигурация сегментации вычислительной сети (ВС) ИС. Защита системных средств – важнейшая составляющая информационной безопасности. Именно на этом уровне необходимо сконцентрировать внимание, поскольку осуществление незаконного доступа к ресурсам системы, приведет к бессмысленности использования прочих мер безопасности;
4. уровень приложений характеризует использование прикладных средств ВС. В связи с тем, что программные средства обеспечивают напрямую работу с различными данными, то для них необходимо использовать собственные средства защиты информации.

Взаимосвязь между уровнями безопасности ИС и уровнями модели OSI отражена на рисунке.

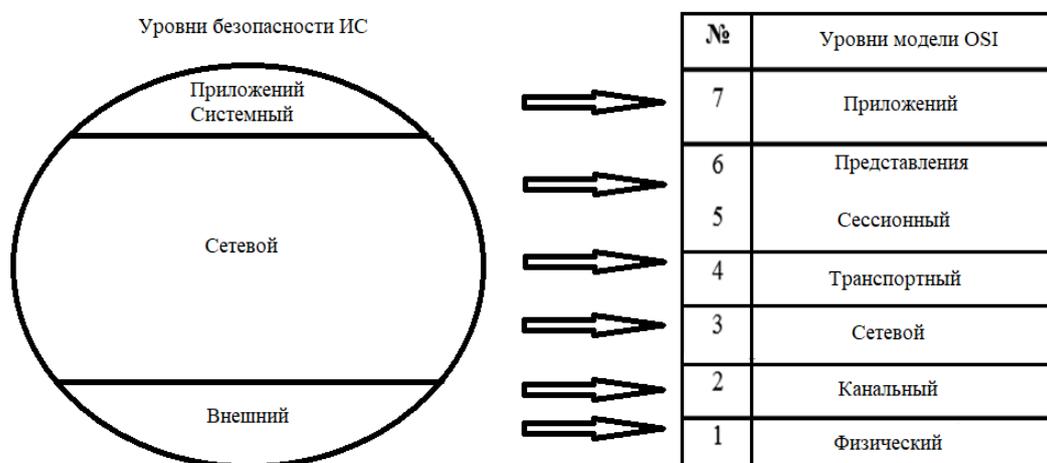


Рисунок. Взаимосвязь между уровнями модели OSI и ИС

Применение различных методик данного подхода обеспечит как качественную, так и количественную оценку возможного информационного ущерба, а также набор процедур, повышающих безопасность интеллектуального имущества предприятия.

Экспертный анализ (ЭА) – процесс, результатом которого является выявление проблемы, полученное исходя из суждения экспертов, на основании которых и будет выбрано последующее необходимое решение. Эти суждения частично выражаются в качественной и частично в количественной форме. ЭА позволяет подготовить информацию для выбора нужного решения лицом, выносящим это решение (ЛВР).

ЭА представляет собой некий интеллект, имеющий более широкие возможности в отличие от отдельного сотрудника. Источником способностей такого интеллекта является выявление ошибочных гипотез и необоснованных мнений, сформированных на основании опыта отдельного сотрудника. С помощью экспертной методики возможно разрешить задачи, нерешаемые аналитической методикой, и такие как:

- выявление наилучшего ситуационного решения из перечня;
- сценарий дальнейшего формирования процесса;
- выбор необходимой методики для решения неоднозначных задач.

Необходимо заметить, что проведение исследования возможно только после четкой постановки цели, а затем формирования надлежащей проблемы в виде вопроса, выносимого на анализ экспертам. Вышеперечисленные действия необходимо производить, руководствуясь следующими правилами:

- необходимо максимально четко установить временные рамки, условия, внутренние и внешние ограничения. Находить решение поставленной задачи, полагаясь на человеческий опыт;
- проблему необходимо преподносить в виде утверждения, но не в качестве численной оценки;
- эксперты выдвигают различные способы решения задачи, не стоит рассчитывать на то, что они предоставят пошаговую схему действий подробного описания каждого выдвинутого решения.

Все имеющиеся экспертные исследования возможно разделить по следующим признакам:

- по типу привлечения экспертов: заочный и очный. Последний тип концентрирует внимание экспертов на поставленной задаче, повышая тем самым качество итоговых решений, в то время как первый тип позволил бы сэкономить финансовые ресурсы;
- по численности повторного проведения анализа (для улучшения показателя точности процедуры): итерационные и разовые;
- в зависимости от решаемой задачи: формирующие решения и анализирующие возможные варианты;
- по виду ответа: концептуальные, ранговые, анализирующие задачу по измерительной шкале;
- по системе обработки экспертных отчетов: прямые и аналитические;
- в зависимости от численности экспертного состава: ограниченные и неограниченные.

Участники экспертного состава должны проанализировать предмет обсуждения с разных точек зрения. Для этого необходимо сформировать и предоставить информационные сведения с подробным описанием объекта, справочные источники, формуляр анкет, необходимые инструменты. Важно не допускать таких ошибок, как: указание авторов материалов, вычленение того или иного способа решения задачи, выражение мнения руководства к итоговым вариантам.

Всю программу анализа проводит независимый куратор, контролирующий выполнение установленных правил, выдающий анкеты и всю необходимую информацию, но не участвующий лично в процедуре.

Выбор экспертного состава – важная и сложная задача в методике экспертной аналитики. Несомненно, суждение каждого участника экспертного состава должно способствовать вынесению адекватного конечного решения, но каким образом выбирать людей на роль экспертов? Определенных способов подбора участников, позволяющих

получить успешную конечную экспертизу, не существует. Вопрос выбора участников можно разделить на две компоненты – формирование перечня возможных членов экспертной комиссии и отбор кандидатур в зависимости от их компетенции. Личная заинтересованность, а также опыт в соответствующей сфере – должны учитываться при подборе участников.

На сегодняшний день разработано огромное количество различных методик выявления экспертного анализа. Некоторые включают в себя работу с каждым экспертом индивидуально, тем самым порождается независимый от авторитетов отчет, так как участник не осознает того, что он является экспертом в данной исследуемой области. Ряд других методик включает четко слаженную работу, позволяя взаимодействовать экспертам между собой, выявляя ошибочные мнения, отбрасывать их. В одних – количество экспертов увеличивается в ходе проведения анализа. В других методиках собрано определенное количество экспертов, учитывая тем самым, согласованность их точек зрения, а затем, усредняя результаты, выносятся обоснованные заключения касаясь принятия нужного решения.

Литература

1. Данелян Т.Я. Формальные методы экспертных оценок // Статистика и экономика. – 2015. – № 1. – С. 183–186.
2. Воеводин В.П. Эволюция понятия и показателей надежности вычислительных систем: Препринт ИФВЭ 2012-24. – Протвино, 2012. – С. 9–12.
3. Кияев В.И., Граничин О.Н. Безопасность информационных систем. – 2-е изд., испр. – М.: Национальный открытый университет «ИНТУИТ», 2016. – С. 12–13.
4. Руденков Н.А., Пролетарский А.В., Смирнова Е.В., Суворов А.М. Технологии защиты информации в компьютерных сетях. – 2-е изд., испр. – М.: Национальный Открытый Университет «ИНТУИТ», 2016. – С. 7–8.
5. Шишкина Н.А. Роль и значение метода экспертных оценок в системе оценивания качества инновационных проектов // Вестник КрасГАУ. – 2013. – № 2. – С. 163.



Алексейцева Елизавета Владимировна

Год рождения: 1996

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: the_dark_one_2033@email.ru

УДК 004.92

**АНАЛИЗ ЗАРУБЕЖНЫХ И ОТЕЧЕСТВЕННЫХ ИССЛЕДОВАНИЙ ПРОБЛЕМЫ
СВОБОДНО РАСПРОСТРАНЯЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ДЛЯ ЦИФРОВОЙ ЖИВОПИСИ**

Алексейцева Е.В.

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 приведен сравнительный анализ отечественных и зарубежных источников по теме «Сравнительный анализ графических редакторов для цифровой живописи». Рассмотрено понятие компьютерной графики и цифровой живописи, поднята проблема отсутствия идеального графического редактора для цифровой живописи.

Ключевые слова: цифровая живопись, графический редактор, концепт-арт, концепт-художник, цифровой художник, компьютерные технологии, художник, дизайнер, Adobe Photoshop, Corel Painter, PaintTool SAI, программа, живопись, рисунок.

Бурный расцвет компьютерной графики приходится на развитие компьютеров и компьютерных технологий. Используя графические редакторы и специализированные программы, предназначенные для создания и обработки цифровых изображений, дизайнеры, художники и программисты разрабатывают концепт-арты, оформление сайтов, компьютерных программ, сотовых телефонов, игровых приставок, и т.д. Многие художники стали обращаться к цифровой живописи, создавая натюрморты, пейзажи, портреты. Работа в графических редакторах открывает новые горизонты для фантазии и творческой деятельности, а также предоставляет творцу широкий спектр возможностей работать в различных направлениях, иметь свой индивидуальный стиль и почерк, основанный на выборе используемых материалов, кистей и эффектов. В основном цифровыми художниками становятся любознательные и энергичные люди, умеющие самостоятельно находить информацию и заниматься самообразованием, дизайнеры, полиграфисты (люди, имеющие опыт работы с компьютерной графикой). Большинство цифровых художников закончили учебные заведения традиционной живописи и рисунка, и впоследствии самостоятельно перешли на CG-арт. Невозможно представить современных цифровых художников без Интернета, где происходит общение с коллегами, поиск работодателей, новых программ, способов рисунка и т.д. Цифровая живопись – изобразительное искусство, создаваемое при помощи компьютерных технологий с применением преимущественно 2D-редакторов. Графические редакторы способны имитировать традиционные техники и материалы, например, масляные или акриловые краски, мягкие графические материалы и т.д. Таким образом, цифровую живопись можно назвать результатом развития и трансформации традиционных видов изобразительного искусства.

Одними из самых распространенных графических редакторов считаются Adobe Photoshop и Corel Painter, PaintTool SAI. Также, в статье Н.А. Малышевой [1] упоминается

такая программа как ArtRage. В сравнении с Adobe Photoshop и Corel Painter, которые являются мощными многозадачными редакторами, ArtRage обладает более простым и удобным интерфейсом, точностью и доступностью. Программа не требует усиленного изучения перед началом работы и проста в использовании. В статье писателя Дейва Жирара [2] также описывается улучшенная и расширенная версия вышеупомянутого графического редактора ArtRage – ArtRage Studio Pro. В преимущества данной программы входят: более совершенное смешение цветов, поддержка фильтра PhotoShop и больше параметров цвета и слоя. Он также обладает отличной функцией экспорта 3D-каналов, которая экспортирует ваше окрашенное изображение в изображение в 3D-рендеринг. Еще одной, наиболее известной программой для цифровой живописи является PaintTool SAI. В статье Е. Головановой [3] данный графический редактор описывается как удобный и быстрый. Еще одним преимуществом выделяется возможность использования данной программы в связке со всем известным Adobe Photoshop. Недостатками автор статьи выделила невозможность рисования автофигур (круги, квадраты, треугольники) в программе и отсутствие работы с текстом. Как и большинство отечественных художников и исследователей, Голованова называет SAI программой, исключительно для работы с мангой, в то время как зарубежные источники не акцентируют на этом внимание. В своем блоге Дж. Чарлтон [4] называет графический редактор SAI легким и дешевым. По мнению Чарлтона, PaintTool удобнее использовать в связке с Photoshop, и рисовать в САИ исключительно лайн, в то время как покрас в том же Adobe Photoshop. Наровне со всеми вышеупомянутыми графическими редакторами, самым известным и распространенным является Adobe Photoshop. А.А. Медведева [5] в своей работе определяет Photoshop в основном как программу для стилизации и обработки фотографий. Данный редактор и правда имеет в своем инструментарии фильтры, имитирующие традиционные художественные техники, такие как: живопись, графика, фреска и тд. Зарубежный график Фил Эбинер [6], как А.А. Медведева, так же рассматривает Photoshop как программу для редакции фотографий и графики. Однако не все писатели и художники расценивают графический редактор Adobe как программу лишь для редактирования готового изображения. И.Е. Голованова и зарубежный художник и блогер Дж. Чарлтон, упоминают Photoshop как инструмент для цифровой живописи. Последней рассмотренной программой для цифровой живописи является Corel Painter. Во многих статьях, как отечественных [1, 3, 7, 8], так и зарубежных [9, 10] авторов, данный редактор упоминается как профессиональная программа, созданная специально для нужд цифровых художников. В рецензионном руководстве к Corel Painter [10], автор называет программу «самой реалистичной в мире», что относится к возможностям Painterа реалистично подражать различным традиционным техникам живописи (акрил, акварель, масло и т.д.), графики и фактурам кистей. Отечественный автор Н.А. Малышева [1], в своей работе пишет, что графический редактор Corel Painter был создан специально для цифровой живописи и рисунка. Также она акцентирует свое внимание на том, что художник добьется большего результата при работе в данной программе, если будет использовать не компьютерную мышь, а графический планшет, что позволит максимально приблизить процесс рисования к привычному традиционному.

На основе проанализированных исследований можно сделать вывод, что на данный момент не существует абсолютно идеального программного обеспечения для цифровых художников. Каждая из рассмотренных программ (Adobe Photoshop, Corel Painter, Paint Tool SAI, ArtRage) имеет как свои «плюсы», так и «минусы», досадным образом осложняющие работу мастера. И отечественные и зарубежные исследователи не выделяют какую-либо программу как самую оптимальную и удобную для цифровой живописи [11, 12].

Литература

1. Малышева Н.А. Цифровые технологии в современном художественном творчестве // Молодежный научно-технический вестник. – 2013. – С. 167–171.

2. ArtRage: quality digital painting on the cheap [Электронный ресурс]. – Режим доступа: <https://arstechnica.com/information-technology/2011/06/artrage-studio-pro-3-review-not-just-the-poor-mans-painter/>, своб.
3. Бугаева П.А. Цифровая живопись как инновация в искусстве // Искусствоведение. – 2016. – С. 11–13.
4. Paint Tool SAI [Электронный ресурс]. – Режим доступа: <http://www.cia.edu/blog/2014/12/paint-tool-sai/>, своб.
5. Медведева А.А. Применение фильтров Adobe Photoshop к цифровой фотографии для имитации традиционных техник живописи // Творчество молодых. – 2016. – С. 110–113.
6. Adobe Photoshop CC: Your Complete Beginner's Guide. Руководство по пользованию [Электронный ресурс]. – Режим доступа: <https://www.adobe.com/learn/courselanding/adobe-photoshop-cc-your-complete-beginners-guide/>, своб.
7. Исаева О.А. Цифровая живопись как актуальное направление отечественного искусства // Вестник СПбГУКИ. – 2017. – № 30. – С. 173–176.
8. Турлюн Л.Н. Специфика художественной выразительности цифровой живописи // Теоретические аспекты искусствознания и культурологии. – 2013. – С. 39–46.
9. How to create a digital oil painting using ArtRage [Электронный ресурс]. – Режим доступа: <http://www.creativebloq.com/how-to/create-a-digital-oil-painting-using-artrage/>, своб.
10. Painter. Reviewer's guide. Руководство пользователя [Электронный ресурс]. – Режим доступа: https://www.painterartist.com/static/ptr/product_content/painter/2018/painter2018-reviewers-guide-en.pdf/, своб.
11. Аккуратора О.Л., Роганова Д.А. Компьютерные технологии в создании концепт-арта // Международная научно-техническая конференция «Информатика и технологии. Инновационные технологии в промышленности и информатике». – 2017. – С. 500–503.
12. Easy PaintTool SAI [Электронный ресурс]. – Режим доступа: <http://mangalectory.ru/lessons/ml1381/>, своб.

**Ананьева Александра Алексеевна**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4205Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: a.a.ananeva@mail.ru

УДК 004.92

ПРОГРАММНЫЕ СРЕДСТВА ВИЗУАЛИЗАЦИИ УПАКОВКИ ТОВАРА**Ананьева А.А.****Научный руководитель – к.т.н. Сокуренько Ю.А.**

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

В работе рассмотрены и проанализированы программные средства визуализации дизайна упаковки товаров. Рассмотрена последовательность работы дизайнера при разработке и создании дизайна упаковки. Описаны все этапы работы, их особенности и наиболее удобные решения задач. Подобраны и описаны популярные программы в работе дизайнера над упаковкой товара.

Ключевые слова: компьютерная графика, дизайн упаковки, разработка упаковки, средства визуализации, компьютерные технологии.

Разработка дизайна упаковки включает в себя несколько важных этапов. В том числе при разработке упаковки очень важно проводить маркетинговые исследования. Также большое значение для оперативной и качественной работы дизайнера имеет выбор программных средств.

На первом этапе разработки дизайна упаковки товара дизайнер совместно с маркетологом, либо самостоятельно определяет концепцию разработки.

В ходе этого этапа должны быть сформулированы предположения о целевой аудитории и потреблении товара, а также должны быть учтены желания потребителей. Наиболее важным шагом является анализ дизайна и концепции продуктов конкурента. Разработка дизайна и поиск концепции происходит с учетом выпущенных ранее продуктов данного бренда. Проводятся исследования эмоционального восприятия товара целевой аудиторией, и производится анализ образа бренда в глазах потребителей [1, 2].

Далее, на втором этапе работы над визуализацией упаковки, дизайнер определяет возможности для визуализации. В ходе работы должен быть подобран ряд ассоциаций и иллюстративный материал для воплощения дизайна. Производится исследование, анализ и подбор цветовых сочетаний и приемов композиции, которые следует использовать для того чтобы передать настроение и эмоциональный посыл производителя. На данном этапе также происходит подбор изобразительных средств для будущей рекламной компании [3].

Третий этап является наиболее важным с технической точки зрения, так как происходит выбор программ для разработки будущей упаковки. В большинстве случаев файлы отправляются на производство в векторном формате. Векторный формат позволяет легко масштабировать объекты и разрабатывать шаблоны для линий штамповки.

Пакет Adobe Creative Suite, включающий программы Adobe Illustrator, InDesign, Photoshop, Acrobat Professional – наиболее популярен для работы над дизайном упаковки. Adobe Creative Suite поддерживает практически все форматы файлов и обеспечивает возможность работать с иллюстрациями, фотографиями и текстом. Приложения, которые

объединены в этой линейке, позволяют работать интуитивно со всеми элементами, поэтому пакет настолько популярен как среди профессионалов, так и среди начинающих дизайнеров.

Создание модели упаковки в 3D-формате дает разработчикам увидеть конечный итог работы перед тем, как запускать упаковку в производство, что позволяет выявить и исправить ошибки, учесть все недостатки и исправить их, используя минимальные затраты и сократив все возможные риски.

Наиболее популярной профессиональной программой для работы с трехмерными объектами является 3D Studio Max. Данная программа позволяет смоделировать и создать любую форму и текстуру упаковки, проработать все детали с высокой точностью. 3D Max имеет широкий спектр возможностей для создания любых, даже самых сложных объектов, позволяет качественно визуализировать дизайн.

Также для визуализации упаковки в трехмерном пространстве часто используется плагин Esko для программы Adobe Illustrator. Данный плагин дает возможность складывать векторный дизайн по линии штамповки и визуализировать его в 3D-формате. Кроме того, позволяет дизайнеру рассмотреть упаковку со всех ракурсов, вращая ее.

Совсем недавно в Adobe Creative Cloud вышел новый продукт для работы с 3D-Dimension CC. Это приложение работает совместно с двухмерной и трехмерной графикой, позволяя создавать фотореалистичные изображения высокого качества, и выполняет композинг 2D- и 3D-ресурсов для визуализации продуктов в реалистичной среде. Для дизайнеров, которые занимаются разработкой упаковки, эта программа наиболее актуальна. Создать реалистичный дизайн возможно простым перетаскиванием графики или логотипа на модель и настройкой материалов. Программа поддерживает интеграцию с Adobe Photoshop, позволяя после экспорта, редактировать любой слой и даже изменять освещение. Вписать 3D-макет упаковки в реальное окружение стало в разы проще. Следует просто выбрать фоновое изображение, а Dimension автоматически устанавливает свет, камеру, перспективу, соотношение сторон. Интерфейс Dimension похож на интерфейсы Photoshop и Illustrator, поэтому дополнительного обучения для дизайнера не потребуется. Для приложения также доступен фотореалистичный рендеринг и предварительный просмотр в реальном времени. Для рендеринга используется V-Ray. Dimension показывает, как будет выглядеть ваш рендер в реальном времени, чтобы получить представление, как будет выглядеть финальная картинка, не тратя время на финальный рендеринг. Для создания материалов доступны новые возможности. Можно создавать материалы, используя Adobe Capture CC, захватывая с его помощью материалы реальных окружающих объектов.

Далее следует четвертый этап работы – создание линии штамповки. Линия штамповки представляет собой плоский шаблон упаковки в векторном формате. Линию штамповки обычно разрабатывают и создают в векторных редакторах, таких как Corel Draw или Adobe Illustrator из простых объектов – линий и фигур. Линия штамповки используется для того, чтобы обозначить, где шаблон должен быть перфорирован (линии сгибов), а где шаблон будет обрезан [4].

На пятом этапе происходит создание дизайна. В начале работы происходит эскизирование и разработка макета будущей упаковки исходя из тех критериев, которые были определены в предыдущих пунктах, в ходе определения целевой аудитории, анализа конкурентов и выбранной концепции.

На данном этапе очень важно учитывать, что дизайн упаковки содержит не только декоративные элементы, но также и дополнительные информативные элементы. Самый главный из них – штрих код. Он обычно располагается на обратной стороне упаковки, и должен быть 16 мм в высоту, исходя из международных стандартов.

Также следует помнить о специальной символике, которая может располагаться на дне упаковки, либо на ее оборотной стороне. В большинстве случаев, это информация о соответствии международным и государственным стандартам, а также о материалах, использованных при производстве [3].

Предпоследний этап – оценка и тестирование дизайна упаковки. Для того чтобы изучить и проанализировать восприятие созданного дизайна целевой аудиторией и сравнить его с упаковкой конкурента, проводится тестирование. Итогом тестирования является определение рекомендаций по доработке дизайна и исправление выявленных несовершенств.

Технологии исследования делятся на качественные и количественные. К первым относятся фокус-группы и глубинные интервью. Эти методы не подходят для тестирования дизайна упаковки, так как вкусы и предпочтения людей довольно субъективны и сильно влияют на результат тестирования. А выборка у этих методов слишком мала и непрезентативна. Количественные методы помогают определить реальную картину восприятия дизайна упаковки целевой аудиторией. Далее производится анализ этих данных. Наиболее эффективные и популярные методы количественных исследований: личное интервью и hall-test [5].

Финальный, седьмой этап разработки – внесение финальных изменений. На этом этапе макет будущей упаковки принимает свой окончательный облик, в соответствии со всеми результатами тестирования.

Литература

1. Борисов Б.Л. Технологии рекламы и PR: учебное пособие. – М.: ФАИР-ПРЕСС, 2001. – 624 с.
2. Моисеева Н.К., Рюмин М.Ю., Слушаенко М.В. Брендинг в управлении маркетингом. – М.: Омега-Л, 2006. – 336 с.
3. Намюр Т. Производство упаковки / Пер. с англ. Дудичев В.М. – М.: Принт Медиа Центр, 2006. – 308 с.
4. Хайн Т. Все об упаковке. – М.: Арт-Родник, 2007. – 288 с.
5. Энджел Д., Блэкуэлл Р., Миниард П. Поведение потребителей. – СПб.: Питер, 2000. – 944 с.



Бабарицкий Павел Александрович

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

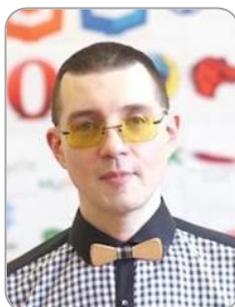
e-mail: redbear95@gmail.com



Сокуренок Юрий Андреевич

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», к.т.н., доцент

e-mail: kpd@limtu.ru



Государев Илья Борисович

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: goss@corp.ifmo.ru

УДК 004.04

**АНАЛИЗ И ИМПЛЕМЕНТАЦИЯ МЕТОДОВ ПАРСИНГА ВЕБ-РЕСУРСОВ. АНАЛИЗ
ЗАРУБЕЖНЫХ И ОТЕЧЕСТВЕННЫХ ИСТОЧНИКОВ ПО ТЕМЕ ИССЛЕДОВАНИЯ**

Бабарицкий П.А., Сокуренок Ю.А., Государев И.Б.

Научный руководитель – к.т.н., доцент Сокуренок Ю.А.

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

В работе описано развитие парсинга, раскрыты причины создания программ для обработки данных, раскрыта значимость их использования. В результате исследования были выделены способы применения парсинга на конкретных примерах, а также рассмотрены методы, которые используются для их создания.

Ключевые слова: парсинг, web minig, web scraping, интеллектуальный анализ данных, синтаксический анализ.

Главной задачей работы было проведение анализа существующих отечественных и зарубежных источников, связанных с методами парсинга веб-ресурсов и их практическим применением. Далее предполагалось на основе результатов проведенного анализа выделить и сформулировать задачи для дальнейшего исследования.

С конца прошлого века возросла значимость накопления, обработки и обмена информации в различных сферах человеческой деятельности. Для решения этих проблем создаются программы, используемые для получения и представления информации, в структурированном формате называемые парсерами.

Одним из мест хранения информации может выступать Интернет.

Еще в середине 90-х годов прошлого века, почти сразу после создания всемирной паутины, в научных работах обсуждалась проблема создания программ для парсинга данных. Так как число информации со временем не перестает увеличиваться, а также появляются различные способы применения парсингов, но при этом отсутствует возможность создания универсального парсера для любой решаемой задачи, то исследование этой проблемы продолжается и в настоящее время, что отражается в современных научных работах.

Для работы с данными используются следующие технологии: базы данных, нейронные сети, машинное обучение, временной анализ и другие.

Со временем появление Всемирной паутины переполнило пользователей домашних компьютеров огромным потоком информации, и чтобы было возможно справиться с обилием доступной информации, ее поиском, сортировкой и фильтрацией, создаются интеллектуальные агенты (softbots). Эти программы создавались такими учеными как, Kozierok и Maes в 1993 году, а также их созданием занимался Oren Etzioni в 1996 году, что отражено в их научных работах [1, 2]. Следует отметить, что Oren Etzioni был первым, кто ввел такое понятие как web mining и своими исследованиями внес значительный вклад в машинное обучение и веб-поиск.

Web mining – это использование методов интеллектуального анализа данных для автоматического обнаружения веб-документов и услуг, извлечения информации из веб-ресурсов и выявления общих закономерностей в Интернете. Также используют понятие web scraping – это программное обеспечение для сбора данных в сети или извлечение веб-данных с веб-сайтов.

Web mining и парсинг можно считать синонимами, но web mining более узкое понятие, применимое для работы с информацией в Интернете.

Синтаксические анализаторы можно разделить на следующие категории в зависимости от того, с какой информацией будет идти работа:

- Web content – обнаружение полезной информации из веб-контента (работа ведется с текстом, изображениями, аудио и видео);
- Web structure mining – анализ ссылок и внешних ссылок сайта (пытается обнаружить модель, лежащую в основе ссылочной структуры в Интернете);
- Web usage mining – интеллектуальный анализ поисковых, серверных или других журналов активности (например, анализ информации о посещении веб-ресурса).

В результате исследования было выяснено, что парсеры применяются для:

- поиска информации в Интернете (это одно из распространенных применений парсеров, среди поисковых систем можно привести в пример Google, Bing, Yahoo!, Яндекс и Mail.ru);
- получения информации об описании товаров и их характеристик для их дальнейшего анализа;
- веб-персонализации (обработки персональных данных пользователей с целью предоставления ссылок, рекламы и рекомендаций), стоит отметить, что это одна из часто поднимаемых проблем среди исследований [3, 4];
- поиск и вывод самой свежей новостной информации;
- разделение веб-ресурсов по возрастным категориям;
- анализа веб-рекламы;
- поиска маркетинговых сведений;
- классификация веб-ресурсов (например, разделение по возрастным категориям);
- проверка уникальности текстов.

Программы получения и представления информации в структурированном формате могут быть написаны программистами на различных языках или сгенерированы специальными утилитами.

Наиболее часто используются такие языки программирования как PHP, Java и Python. В последние годы среди иностранной литературы появляются публикации, посвященные разработке парсеров на этих языках [5, 6].

Среди утилит можно привести в пример программы Xrefer и Xcovator. При использовании этих и подобных программ парсинг будет идти в зависимости от выбранных параметров, предусмотренных программой.

В процессе исследования значительного числа отечественных и иностранных источников:

- было выявлено общее представление о применении парсинга, в сфере веб-технологий и методах его реализации;
- обобщены отечественные и зарубежные исследования, посвященные теме обработки информации с использованием синтаксических анализаторов;
- выделены наиболее актуальные научные задачи, требующие дальнейшего исследования, в том числе и задача анализа и имплементации методов парсинга веб-ресурсов.

Литература

1. Kozierek R. Maes P. Learning interface agents // Proceedings of the 11th National Conference on Artificial Intelligence (AAAI-93). – 1993. – P. 459–465.
2. Etzioni O. World-Wide Web: Quagmire or gold mine // Communications of the ACM. – 1996. – № 39(11). – P. 65–68.
3. Mobasher B. Data mining for web personalization // The adaptive web. – 2007. – P. 90–135.
4. Mobasher B., Dai H., Luo T., Sun Y., Zhu J. Integrating Web Usage and Content Mining for More Effective Personalization // Electronic commerce and web technologies. – 2000. – P. 165–176.
5. Ward J. Instant PHP Web Scraping. – Birmingham: Packt Publishing, 2013. – 60 p.
6. Sweigart A. Automate the Boring Stuff with Python: Practical Programming for Total Beginners. – San Francisco: No Starch Press, 2015. – 504 p.

**Бабахин Антон Валерьевич**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4108Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: anton-098@yandex.ru

УДК 004.5

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПОДХОДОВ К РАЗРАБОТКЕ ВЕБ-ИНТЕРФЕЙСОВ**Бабахин А.В.****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 определены основные проблемы, возникающие при разработке пользовательских веб-интерфейсов. Рассмотрены основные подходы к эффективной разработке веб-интерфейсов. Выявлены положительные и отрицательные стороны каждого из подходов. Сделан вывод о том, какой подход следует применять при определенных требованиях к проекту.

Ключевые слова: веб-интерфейс, Twitter Bootstrap, Foundation, Kendo UI, CSS-методологии, ВЕМ, OOCSS, SMACSS.

На сегодняшний день сфера интернет-технологий демонстрирует один из самых больших темпов роста по сравнению с другими отраслями экономики.

Ключевую роль при взаимодействии пользователя с ресурсом в сети Интернет играет интерфейс. Ведь понятный, легкий, красивый интерфейс позволяет пользователям комфортно взаимодействовать со страницами сайта, что увеличивает их лояльность к компании.

Одной из проблем при разработке интерфейсов сайтов является большое количество разных браузеров, через которые пользователи смотрят страницы. Разные браузеры могут поддерживать разные веб-технологии, ведь на текущий момент стандарты этих технологий не являются обязательными к внедрению.

Кроме этого, значительно отличаются размеры экранов устройств для просмотра страниц сайта. Пользователь может открыть веб-ресурс как на мобильном телефоне, так и на персональном компьютере.

Еще одной проблемой разработки пользовательского интерфейса для веб является возможное увеличение команды разработчиков, работающих над одним проектом. У каждого разработчика свой стиль написания кода, свои способы решения определенных задач. Сотрудник может использовать те инструменты, которые другой сотрудник не использует или не знает.

Один из распространенных подходов к реализации пользовательских веб-интерфейсов – использование библиотек. Они предоставляют разработчику большой набор готовых элементов интерфейса: формы, кнопки, меню, цвета и т.д. К тому же подобное решение позволяет делать интерфейс отзывчивым, подстраивающимся под размеры конкретного экрана. Сегодня библиотека Bootstrap компании Twitter является самым популярным фреймворком такого рода. Библиотека имеет специальную систему сетки, облегчающую разработку веб-интерфейсов под устройства различных форм-факторов. Разработчику требуется с помощью специальных классов определить расположение элементов на экране смартфона, планшета или монитора персонального компьютера.

Библиотека содержит набор предопределенных CSS-классов, что позволяет не задумываться о внешнем виде элементов на начальном этапе разработки, так как в результате получается уже стильный и современный макет интерфейса [1]. Помимо Bootstrap, существуют другие библиотеки для реализации пользовательских интерфейсов, например, Foundation или Kendo UI.

Главная трудность при использовании HTML/CSS фреймворков наподобие Bootstrap и Foundation – редизайн. Данные технологии используют строки и столбцы для проектирования интерфейса, поэтому невозможно провести редизайн сайта просто внося изменения в CSS. Придется менять структуру колонок и строк.

Также необходимо отметить, что отрицательной стороной данного подхода является увеличение времени начальной загрузки интерфейса. Дизайн оказывается тяжеловесным, приложение становится полностью зависимо от выбранного фреймворка [2].

Другой подход к разработке пользовательских интерфейсов – применение CSS-методологий. CSS-методологии – это наборы правил к организации кода, файловой структуре проекта и используемому программному обеспечению.

Сегодня существует большое количество различных методологий:

- BEM;
- SMACSS;
- OOCSS;
- MCSS;
- OPOR;
- Atomic CSS;
- AMCSS;
- DoCSSa;
- FUN.

Исследователи отмечают, что методологии OPOR и FUN могут быть использованы для создания небольших и одностраничных сайтов, так как обладают небольшим количеством правил, не требующих жесткого исполнения, а также не привязаны к конкретным инструментам [3].

BEM-методология является одной из самых популярных методологий верстки, существующих на сегодняшний день. Она позволяет поддерживать проекты на протяжении всего жизненного цикла, снижать к минимуму лишнюю работу разработчиков, позволяет эффективно использовать CSS-препроцессоры при работе с ней [4]. Отрицательной стороной подхода является часто возникающие излишне длинные имена классов.

Методология OOCSS позволяет максимально оптимизировать код для браузеров. Также над проектом может работать большая команда разработчиков. В качестве недостатков метода можно отметить, что при слишком большом количестве объектов в проекте можно забыть, какие объекты уже были созданы, и сделать их дубликаты.

Плюсом SMACSS является то, что данная методология имеет правила для определения базовых HTML-элементов, позволяет описывать состояния элементов, разрешено использование ID для базовых элементов. В SMACSS стили разделены по 5 слоям – базовый, стили слоя, стили модулей, стили состояний и стили темы.

Многие исследователи отмечают, что самый правильный подход к организации разработки интерфейса – гибрид методологий БЭМ и SMACSS. Можно использовать три уровня SMACSS – base, layout и modules, а также принцип именования элементов BEM [5].

Таким образом, можно сделать вывод, что для ускорения процесса разработки пользовательских веб-интерфейсов можно использовать готовые библиотеки элементов. Но объем страниц в данном случае увеличивается, что сказывается на времени их отрисовки.

Использование CSS-методологий верстки является наиболее универсальным способом создания интерфейса, так как позволяет создавать легко масштабируемый и модульный код. БЭМ-методология является одной из самых популярных, но у нее также есть и определенные

недостатки. Исходя из этого, для выбора методологии для конкретного проекта нужно внимательно изучать их положительные и отрицательные стороны.

В ходе дальнейших исследований планируется провести подробный сравнительный анализ CSS-методологий с определением критериев сравнения.

Литература

1. Дмитриевский С.В. Современные методы прототипирования графического пользовательского интерфейса для web-ориентированной среды // Актуальные вопросы современной науки. – 2015. – № 3. – С. 12–16.
2. Ушаков М.В. Меньше кода – больше дохода. Часть 4. Разработка front-end-слоя // Системный администратор. – 2014. – № 6. – С. 78–79.
3. Селиванова Е.А., Левина И.В. Методологии верстки: БЭМ, АМССС, ООСС, Atomic CSS, ОРОР, МССС, SМАССС, FUN, DoCSSa [Электронный ресурс]. – Режим доступа: <http://html5.by/blog/bem-amcss-oooss-atomiccss-opor-mcss-smacss-fun-docssa-video/>, своб.
4. Методология / БЭМ [Электронный ресурс]. – Режим доступа: <https://ru.bem.info/methodology/>, своб.
5. Catanzariti P. BEM and SМАССС: Advice From Developers Who've Been There [Электронный ресурс]. – Режим доступа: <https://www.sitepoint.com/bem-smacss-advice-from-developers/>, своб.



Баев Андрей Александрович

Год рождения: 1996

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: ghost9610@yandex.ru



Государев Илья Борисович

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: goss@corp.ifmo.ru

УДК 004.02

ОБЗОР ИССЛЕДОВАНИЙ ПО ПРОБЛЕМЕ ПРОЕКТИРОВАНИЯ МНОГОПОЛЬЗОВАТЕЛЬСКИХ ИГР НА ВЕБ-ПЛАТФОРМЕ

Баев А.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 проведен обзор исследований по проблеме проектирования игр. Рассмотрены основные составляющие разработки. Выделены основные проблемы и вопросы, возникающие в процессе разработки.

Ключевые слова: компьютерные игры, разработка игр, геймдизайн, проблемы проектирования игр, информационные технологии.

На сегодняшний день уровень развития компьютерных технологий позволяет нам использовать их во многих сферах деятельности, в том числе и в развлечениях. Одним из таких направлений являются компьютерные игры. Компьютерные игры получили свое начало во второй половине двадцатого века, и продолжают развиваться в настоящее время.

Несмотря на развитие технологий, проблема проектирования игры, которая понравилась бы огромному количеству людей, все также актуальна.

Эта актуальность объясняется тем, что с ростом технологий и улучшением качества предложения, растет также и спрос на новые виды развлечений, на еще большее повышения их качества.

Помимо роста качества предложения, открываются и новые направления компьютерных игр, которые, в свою очередь, тоже требуют исследований и планирований проектирования. Одним из таких примеров может быть недавно появившейся термин VR (Virtual Reality) – игры с виртуальной реальность, или термин AR (Augmented Reality) – игры с дополненной реальностью.

Не вызывает сомнений, что компьютерные игры действительно одна из важных ветвей IT-технологий, спрос на которую будет только усиливаться. Так, например, известная компания Newzoo, чьи исследования заказывают такие гиганты игровой индустрии как Valve, King, Sega, Wargaming, Electronic Arts, провела исследование [1, С. 1], согласно

которому с 2015 по 2017 год аудитория видеоигр выросла с 1,2 до 2 млрд человек. Newzoo отмечает, что в будущем спрос на игры будет только увеличиваться.

Перед проектированием любой игры следует определить цель создания игры, определить жанр, setting (обстановку, атмосферу, окружающую среду) и ответить на вопрос – какая игра будет создана? Не менее важно определиться какая команда будет создавать игру – она может варьировать от одного, двух человек, до масштабов целой студии.

Также следует определить и инструменты разработки, которые будут использоваться при создании игры. Следует обратить особое внимание на их лицензионные соглашения.

Одним из немаловажных вопросов является и вопрос о предполагаемых средствах, которые будут потрачены на разработку и последующую поддержку игры.

Этап проектирования игры можно назвать геймдизайн.

В книге авторов Brenda Brathwaite и Ian Schreiber, «Challenges for game designers», описано подробно, что из себя представляет геймдизайн [2, С. 2].

Геймдизайн – процесс создания игрового контента и правил. Хороший геймдизайн – процесс создания целей, которые игрок захочет достигнуть, и правил, которым игрок будет следовать в процессе принятия значимых решений на пути к достижению этих целей.

При создании игры геймдизайнер должен попытаться увидеть игру глазами игрока, чтобы понять ответы на простые вопросы:

- О чем эта игра?
- Как я играю?
- Как я побеждаю?
- Почему я хочу играть?
- Что мне нужно делать в игре?

По мнениям авторов, геймдизайн можно считать процессом создания для игрока набором выборов, которые повлияют на исход игры. Причем выбор того или иного действия может как приблизить игрока к победе так и наоборот отдалить. Их может быть много. Если игрок играет в real-time стратегию, то решений, которые надо принимать становится более нескольких десятков в минуту.

Создать выбор можно, придумав альтернативное сколь угодно значимое решение задачи. Не всегда есть выбор. И игрокам это не нравится.

Авторы книги отмечают также, что существует множество видов геймдизайна. В него входят:

- системный дизайн – создание правил и сопутствующих расчетов для игры. Это этап, когда проектируется будущая «математическая» логика, игровая логика. У всех игр есть своя игровая логика, по которой и происходит взаимодействие игрока с игрой;
- дизайн игровой среды – создание темы, атмосферы и внутренней истории игры. От дизайна игровой среды зависит все остальное проектирование. Основной облик игры так же зависит от этого этапа;
- контент-дизайн – создание наполнения игры. Сюда входят различные игровые сущности, с которыми будет взаимодействовать игрок;
- игровые тексты – это написание внутриигровых диалогов, текстов и историй;
- дизайн уровней – создание уровней игры, включающей ландшафт карты и расположение на этой карте объектов;
- дизайн интерфейсов (UI) – можно назвать совокупность того, как игрок взаимодействует с игрой, и как игрок получает информацию и реакцию на свои действия от игры. Интерфейс должен быть интуитивно понятен для игрока, ведь от информации с него будут зависеть дальнейшие события, и общее чувство удовлетворения игрой.

Рассмотрим системный дизайн. Как отмечалось выше, это этап разработки логики взаимодействия игрока с игровой средой. Это взаимодействие принято называть геймплеем.

В работе Emmanuel Guardiola, «The Gameplay Loop: A Player Activity Model for Game Design and Analysis», автор рассматривает принцип проектирования главного элемента геймплея [3]. Он называется *gameplay loop* или игровой цикл.

Игровой цикл выступает в качестве каркаса, к которому могут добавляться некоторая вариативность взаимодействия игрока с игрой. Цикл повторяется в той или иной степени на протяжении всей игры. Иногда в более сложных играх, игровых циклов может быть несколько, например, в игре «Sid Meier's Civilization», и они могут идти как параллельно, так и последовательно.

Один из примеров цикла продемонстрирован на рисунке, а.

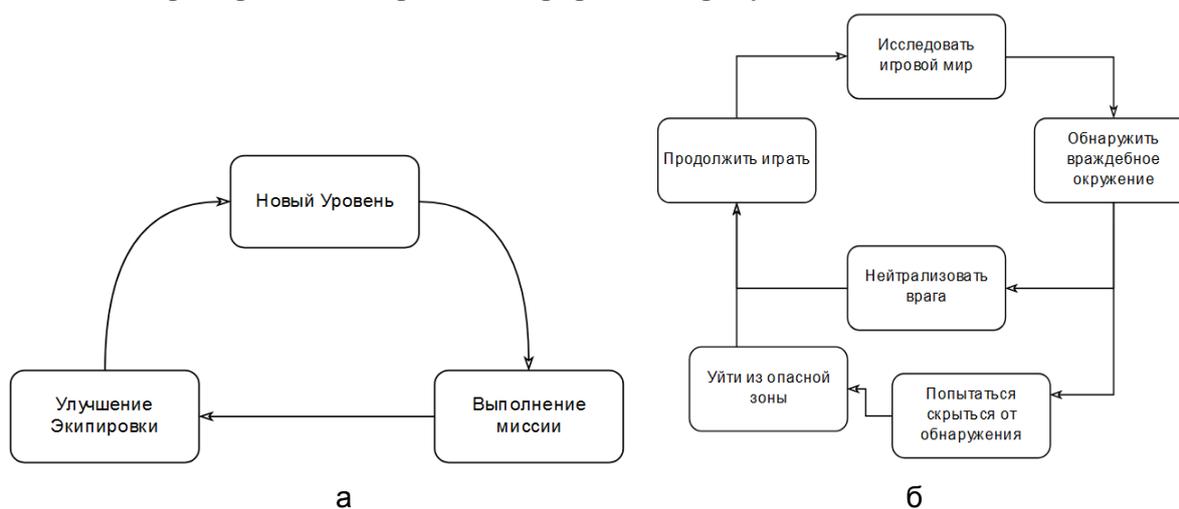


Рисунок. Пример игрового цикла (а) и игрового цикла с вариативностью (б)

Как показано на рисунке, а, основой игровой механики составляет выполнения миссий, после которых игрок получает возможность улучшать свою экипировку, или какие-либо игровые значения игрока, здесь все зависит от конкретных проектов. И наконец, открывается новый уровень, как новый вызов игроку, и тут происходит заикливание геймплея – игроку снова нужно выполнять миссии и снова идти по этому циклу.

На рисунке, б, продемонстрирован игровой цикл с вариативностью действий игрока. Допустим в игре при исследовании мира можно обнаружить враждебное окружение или врагов, тогда игроку дается выбор, пойти путем борьбы или уйти подальше от опасностей.

Emmanuel Guardiola в своей работе отмечает, что при использовании такого подхода разрабатывать базовые механики и базовый геймплей, на ранних этапах разработки проекта, становится проще. Уже позже автор отмечает, куда легче вносить различные вариативности.

Tynan Sylvester в своей книге «Designing Games: A Guide to Engineering Experiences» [4], считает, что геймдизайн – это не то, чему можно научиться прочитав книги, это то, что требует практического опыта. Автор также отмечает, что хорошее изучение геймдизайна происходит через наблюдения результатов маленьких, изолированных изменений в игровой логике.

Описывая контент-дизайн и дизайн игровой среды, Tynan отмечает, что повествование – это то, что формирует историю в разуме игрока.

Jesse Schell в своей книге «The Art of Game Design: A Book of Lenses», отмечает, что главное в геймдизайне это умение слушать [5, С. 6]. Слушать команду, аудиторию будущей игры, игру, себя.

- геймдизайнер должен слушать команду, ведь вместе они создают игру и принимают важные решения;
- он должен слушать аудиторию будущей игры, ведь именно для них она создается. Если им не понравится играть в игру, то дизайн не удался;

- геймдизайнер должен слушать игру, он должен знать ее «вдоль и поперек», автор приводит в пример механика, который может рассказать о поломке по звуку двигателя машины;
- и, наконец, геймдизайнер должен слушать себя. Он должен понимать, чего хочет добиться, представлять каждую деталь будущей игры, выстраивать план в голове.

Игра – это развлечение. Разработчик должен определить те элементы игры, которые должны служить развлечению игрока и быть для этого игрока значимыми. Геймдизайнер может попробовать сделать это более значимым, понять, как мотивировать игрока. Из этого можно предположить, что следует изучить аудиторию своей будущей игры, чтобы ответить на эти вопросы, чтобы провести анализ и подкорректировать свои наработки.

Автор поднимает вопрос о проблемах, решаемых в играх. Он советует подумать при проектировании над тем, что должен будет сделать игрок на протяжении всей игры, или как игра будет генерировать новые задачи, чтобы игрок мог возвращаться и играть снова. Возникает вопрос о так называемом термине «high-level gameplay». При создании некоторых игр, например, мультиплеерных MMORGP (Massively multiplayer online role-playing game), следует задумываться о том, что будет удерживать игрока в игре на высоких уровнях, будут ли занятия для игрока, который практически всего добился в игре.

Литература

1. Takahashi D. Global game business is booming [Электронный ресурс]. – Режим доступа: <https://venturebeat.com/2014/06/03/global-game-business-is-booming-but-physical-store-sales-may-evaporate-by-2015/> (дата обращения: 20.11.2017).
2. Brathwaite B., Schreiber I. Challenges for game designers [Электронный ресурс]. – Режим доступа: <http://www.quilgeo.com/wp-content/uploads/2013/07/Challenges-for-Game-Designers.pdf> (дата обращения: 20.11.2017).
3. Guardiola E. The Gameplay Loop: A Player Activity Model for Game Design and Analysis. – New York, USA: ACM, 2016. – С. 23:1–23:7.
4. Sylvester T. Designing Games: A Guide to Engineering Experiences [Электронный ресурс]. – Режим доступа: <http://www.reedbushey.com/32Designing%20Games.pdf> (дата обращения: 10.12.2017).
5. 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: 10.12.2017).



Базаева Кристина Алановна

Год рождения: 1996

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4106

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: kristinabaz@bk.ru

УДК 004

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МУЗЫКАЛЬНОМ ОБРАЗОВАНИИ

Базаева К.А.

Научный руководитель – д.т.н., профессор Готская И.Б.

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

В работе рассмотрены цели и способы использования информационных технологий в музыкальном образовании.

Ключевые слова: информационные технологии, технологии в музыкальном образовании, игровые технологии, музыка.

На сегодняшний день информационные технологии охватывают практически все сферы деятельности, не обошли они стороной и музыку. На самом деле информационные технологии уже давно играют важную роль в музыкальном творчестве. В наши дни сложно представить, как люди раньше обходились без компьютеров и специального программного обеспечения, позволяющего создавать композиции, не имея при этом даже реальных музыкальных инструментов, а также других полезных программных продуктов, предназначенных для музыкантов.

Кроме того, современные технологии применяются и в музыкальном образовании. Существует множество программ и приложений, позволяющих обучаться музыке.

В настоящее время в музыкальной деятельности, в том числе в образовательных целях, используются как персональные компьютеры, так и планшеты и смартфоны.

Мобильные телефоны давно перестали быть просто средством связи. Современные смартфоны имеют множество функций, они используются для прослушивания музыки, создания заметок, документов, фотографий, а также доступа к Интернету и установки различных приложений. По сути, смартфоны – это компактная копия компьютера.

При этом в наше время смартфоны и планшеты используются людьми даже чаще, чем ноутбуки или персональные компьютеры. Именно поэтому появляется все больше обучающих приложений для смартфонов и планшетов, которыми пользуются тысячи людей.

В обучении с помощью компьютерных программ есть свои особенности:

1. использование игровых технологий. С их помощью можно обучаться и сразу видеть свой прогресс. За каждый пройденный этап пользователь получает виртуальные награды, что повышает его мотивацию;
2. отсутствие фиксированного графика. Пользователь сам выбирает удобное время для занятий;
3. цена. Обучение с помощью приложений обойдется гораздо дешевле, чем занятия с репетиторами или специальные курсы, кроме того, существует множество бесплатных обучающих приложений.

В настоящее время целями использования технологий в музыкальном образовании являются: повышение мотивации у обучающегося; развитие научного мышления обучающегося; расширение методов обучения; развитие навыков самообразования.

К основным направлениям обучения музыке с использованием современных обучающих приложений можно отнести:

- получение теоретических знаний. Для того чтобы учащиеся усвоили основы и правила построения музыкальных произведений используются игровые технологии для запоминания специальных терминов, нот, интервалов и т.д.;
- техническое мастерство. Существуют программы, помогающие овладеть базовыми навыками игры на выбранном пользователем инструменте;
- развитие слуха. Приложения включают в себя различные слуховые диктанты, с помощью которых обучающийся может тренироваться в определении нот, аккордов, гамм и т.д., а также упражнения с графической фиксацией высоты тона для контроля чистоты вокальной интонации;
- музыкальная литература – некоторые программы представляют собой учебные пособия со звуковыми файлами различных музыкальных произведений, краткие биографии соответствующих композиторов и музыкантов, небольшой тест, основанный на представленном материале. Другие приложения позволяют пользователю оценивать свои знания об истории и теории музыки. Обучающиеся также выполняют задания, такие как прослушивание, чтение, запись и музыкальная форма или гармонический анализ. Эти программы способны обеспечить более постоянное обучение, поскольку они позволяют тренироваться на любом желаемом уровне;
- развитие креативности – программы, предназначенные для улучшения навыков в области композиции и аранжировки [1–3].

С помощью информационных технологий можно создать эффективные тренажеры для любых дисциплин в музыкальном образовании. Использование технологий дает возможность развивать музыкальные способности у человека, что обусловлено их интенсивными обучающими возможностями, а также автоматизировать практические занятия по развитию слуха, навыков в области композиции и аранжировки, чтению нот и т.д. [1]. Кроме того, с использованием технологий намного проще составлять план обучения и вести контроль за усвоением пройденного материала.

Однако необходимо подчеркнуть, что роль учителя музыки в обучении с использованием информационных технологий не уменьшается, более того самообразование может привести к неправильному обучению.

Таким образом, технологии довольно широко используются в музыке. Хорошо продуманные программы и приложения могут быть полезным дополнительным инструментом в музыкальном образовании.

Литература

1. Горбунова И.Б. Феномен музыкально-компьютерных технологий как новая образовательная творческая среда // Изв. Рос. гос. пед. ун-та им. А.И. Герцена. – 2004. – № 4(9). – С. 123–138.
2. Горбунова И.Б., Романенко Л.Ю., Родионов П.Д. Музыкально-компьютерные технологии в формировании информационной компетентности современного музыканта // Науч.-техн. вед. СПбГУ. Гуманит. и обществ. науки. – 2013. – № 1(167). – С. 39–48.
3. Nilsson B., Folkestad G. Children's practice of computer-based composition // Music Education Research. – 2005. – № 7. – P. 21–37.



Бачурина Людмила Романовна

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4205

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: lusya.bachurina@gmail.com

УДК 004.9

АНАЛИЗ СТРУКТУРЫ ДИНАМИЧЕСКИХ ВИЗУАЛЬНЫХ СИСТЕМ

Бачурина Л.Р.

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 рассмотрены особенности формирования целостной динамической системы идентичности бренда, основные ее компоненты, способы организации связей между ними.

Ключевые слова: динамическая айдентика, визуальные системы, генеративный дизайн, параметрические методы проектирования, алгоритмический дизайн.

Интенсивное развитие средств коммуникаций, появление новых каналов взаимодействия непосредственно влияют на инструменты общения, что способствует формированию нового понимания терминов «айдентика», «бренд», «фирменный стиль». Разработка айдентики на сегодняшний день все чаще подразумевает создание визуального образа, наделенного механизмами взаимодействия с пользователем и трансформаций для представления в различном контексте и на различных видах носителей, т.е. проектирование динамической визуальной системы. Расширение спектра возможностей программного обеспечения, развитие технологий генеративного дизайна позволяют реализовывать особенно сложные системы идентичности и не только повысить эффективность и гибкость брендинга, но и решить ряд практических задач: визуализация больших объемов данных в режиме реального времени, визуальное отражения многопрофильности деятельности компании, генерация вариативных компонентов системы, автоматическая персонализация в рамках единого стиля на многопользовательских платформах. Разработка динамической айдентики включает создание набора вариативных элементов, объединенных единым стилем и идеей, отражающей концепцию компании. Таким образом, возникает проблема обеспечения целостности и узнаваемости идентификаторов бренда при различных внешних и внутренних изменениях. Также система должна быть представлена на математическом языке, т.е. адаптирована для реализации динамичности программными средствами с помощью генеративных технологий и алгоритмических методов. Данные факторы обуславливают необходимость изучения элементов визуальных динамических систем, процесса формирования связей между ними и особенностей их зрительного восприятия.

В отличие от традиционного фирменного стиля, который включает в себя ограниченный набор статичных компонентов, визуальная динамическая система формируется из более широкого и адаптивного набора компонентов. Ирен ван Нес в работе «Динамическая айдентика. Как создать живой бренд» [1] определяет следующие составляющие структуры системы динамической визуальной идентичности: логотип, цвет, типографика, графические элементы, образ и язык бренда. Каждый компонент уточняет личность бренда, который он представляет. Чем больше компонентов определено, тем легче идентифицируется система. Данные компоненты, объединенные в систему, формируют

идентичность бренда. Система идентичности является динамической, когда один или несколько компонентов определяются, как доступные для вариаций. На рис. 1 представлены системы визуальной идентичности бренда с динамическими (отмечены белым) и статическими (отмечены синим) компонентами. Варьирование одного из элементов приводит к трансформации связей, однако система сохраняет целостность. Очевидно, что чем больше компонентов определено и чем больше из них являются вариативными, тем более сложной задачей становится сохранение целостности восприятия системы.

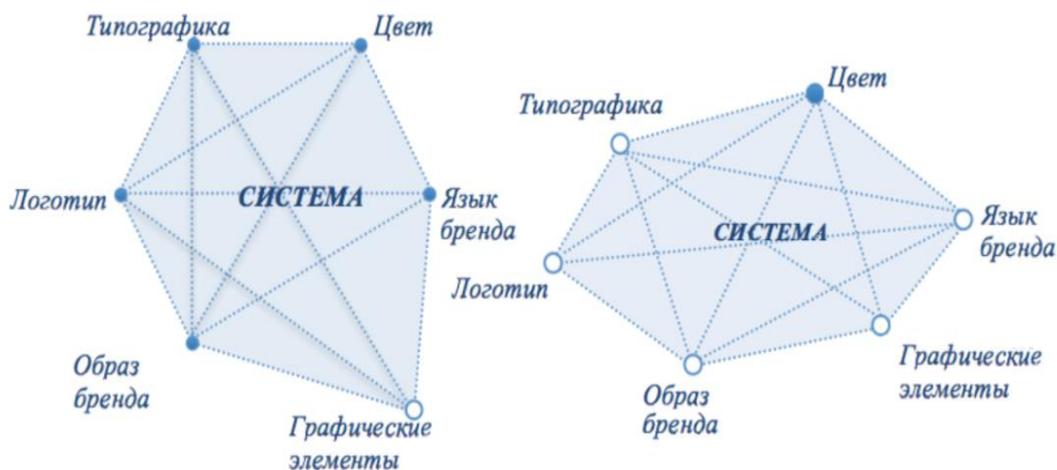


Рис. 1. Системы визуальной идентичности бренда с динамическими компонентами

Определение и разработка компонентов динамической визуальной идентичности бренда, а также организация связей между ними является основой проектирования системы. Однако необходимо также определить принципы их взаимодействия. Решению данной задачи способствуют модели систем динамической айдентики, разработанные Ирен ван Нес [1]: «Контейнер» (Container), «Обои/фон» (Wallpaper), «ДНК» (DNA), «Формула» (Formula), «Кастомизация» (Customised), «Генеративная» модель (Generative). Каждая модель описывает принцип осуществления гибкости динамической визуальной идентичности посредством различных способов конфигураций ее компонентов. Данная концепция вызвала интерес у исследователей и подробно рассмотрена в научных работах Э. Джошума (Emanuel Jochum, «Dynamic Branding Thesis») [2], [3] и других, что обуславливает ее значимость для понимания принципов формирования динамических систем идентичности.

Наименее контролируемой дизайнером, но наиболее гибкой и многообразной является «Генеративная» модель, которая основана на генерации идентификаторов с помощью инструментов программирования. В качестве входных параметров для такой системы, как правило, выступают реальные внешние данные (количество, температура, скорость, направление движения и т.д.). Программными методами данные трансформируются в динамические графические образы, являющиеся ядром визуальной идентичности. Для ее реализации модели необходимо решить две основные задачи: обеспечить единство восприятия системы и представить структуру динамической визуальной системы не только на уровне идеи и концепции, но и в геометрическом и параметрическом виде для применения средств программирования.

Как правило, динамическая визуальная система идентичности является достаточно сложной и объединяет множество компонентов и их вариаций, поэтому работать с ней даже на уровне концепции достаточно сложно. Еще более сложной становится задача написания руководства и представления структуры в программном виде. В данном случае эффективно применение «Общего метода для не количественного моделирования» или метода «Морфологический анализ» (General Morphological Analysis (GMA)) [4]. Морфологический анализ был разработан в 1940-х годах швейцарским физиком Ф. Цвики и впервые был представлен в графическом дизайне в 1964 году К. Герстнером, разработавшим

морфологическую коробку Цвикки («Zwicky Morphological box»). К. Герстнер видел морфологический анализ как механизм перестановки, полезный для быстрого и эффективного создания большого количества прототипов решений для проектов, связанных с визуальной идентификацией [3, 5]. Суть метода морфологического анализа заключается в построении матрицы параметров будущего продукта и включает следующие этапы: постановка задачи, определение набора базовых характеристик объекта, заполнение матрицы вариантами параметров базовых характеристик, генерация комбинаций характеристик объекта из элементов матрицы, обработка результатов и адаптация под конкретную задачу, техническое задание. Поскольку динамические системы идентичности предполагают большое число вариативных элементов, данный метод позволяет охватить наибольшее число возможных изменений параметров идентификаторов на идейном уровне, а также сформировать набор входных данных для реализации проектов алгоритмического дизайна, определить направления трансформаций всей визуальной системы в совокупности или ее составных частей. При этом разработка собственного морфологического поля позволяет получить не стандартные решения (например, динамика цвета, переход формы), а индивидуальные, присущие конкретной системе и специфическим параметрам, что способствует разработке уникальной айдентики.

Еще одной важной задачей проектирования является определение границ изменений вариативных компонентов. Для применения возможностей алгоритмического дизайна, структура динамической визуальной системы идентичности должна базироваться на математических принципах, что позволит трансформировать ее в формат, доступный для дальнейшего программирования. Решению данной задачи способствует концепция «проектных осей» Г. Норджея, которая позволяет строить проектное пространство, на основе декартовой системы координат, содержащее множество вариаций объекта, параметры которого эквивалентны одной из проектных осей. На рис. 2 представлена модель Г. Норджея на примере типографского символа «е», демонстрирующая результаты изменения объекта по трем параметрам (осям), выделены допустимые границы вариаций, в которых символ остается узнаваемым. Элементы в узлах куба являются узлами интерполяции (ключевыми формами) и их изменение влияет на весь диапазон вариаций. Параметрами могут быть цвет, толщина линий, угол поворота и т.д. Также модель может являться одноосевой, т.е. представлена прямой линией от «А» до «В», с мастер-дизайном (ключевой вариацией элемента) в каждой конечной точке.

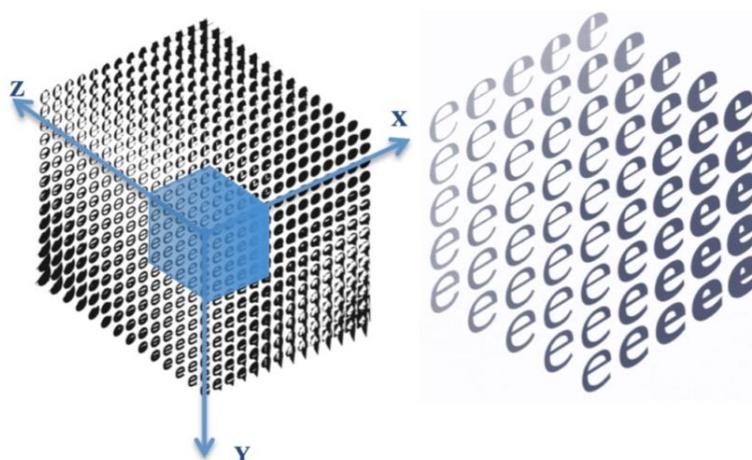


Рис. 2. Модель Г. Норджея и результат интерполяции

Основным научным интересом Г. Норджея являлась типографика, и его концепция осей была разработана для решения задач этой области. В дальнейшем понимание концепции Г. Норджея было расширено, и идея нашла применение в проектировании логотипов и создании других компонентов систем идентичности. Интерполяция, лежащая в ее основе,

является методом построения новых элементов в пределах диапазона дискретного набора известных точек данных, т.е. осуществляется плавный переход одной формы в другую. Данный процесс проектирования очень гибкий и временные затраты на поиск и трансформацию форм значительно снижаются. Каждый элемент можно точно настроить, контролируя интерполированные экземпляры. Очевидно, что при необходимости работы с большими объемами данных, вручную выполнить интерполяцию невозможно, процесс осуществляется программными методами. В таком случае дизайнеру необходимо четко определить статические и динамические оси, определив набор ключевых точек и параметров.

Таким образом, для формирования целостной динамической систем идентичности и однозначной идентификации бренда в различных контекстах на этапе проектирования структуры системы необходимо сохранять баланс между статическими и динамическими компонентами, обеспечивать «бесшовность» трансформаций динамических компонентов за счет выбора схожих узлов интерполяции и создания на их основе континуума промежуточных изменений в пределах определенного вариационного спектра. Также система должна быть представлена на математическом языке, т.е. адаптирована для реализации динамичности программными средствами с помощью генеративных технологий и алгоритмических методов.

Литература

1. Nes I., *Dynamic Identities: How to Create a Living Brand* // BIS Publishers. – 2013. – 192 p.
2. Jochum E. *How flexible design systems turn brands into dynamic visual identities* // Master's thesis. – 2013.
3. Junior S., Alves J. *Identidades visuais flexiveis: das origens ao projeto* // Master's thesis. – 2015.
4. Ritchey T. *General Morphological Analysis – A general method for non-quantified modeling*. – Springer-Verlag, 2002. – 3 p.
5. Gerstner K. *Designing Programmes*. – Netherlands, 1964. – 256 p.



Башкирцева Екатерина Ивановна

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4205

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: b-ka95@mail.ru

УДК 004.9

ОБЗОР РЫНКА ПРИЛОЖЕНИЙ И УСТРОЙСТВ В СФЕРЕ MHEALTH

Башкирцева Е.И.

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 проведен обзор рынка приложений и устройств в сфере mHealth. Дано пояснение об актуальности мобильной медицины в современном мире. Раскрыты основные понятия mHealth, описаны наиболее популярные устройства и приложения в области контроля физиологического состояния.

Ключевые слова: mHealth, мобильные приложения, устройства сферы mHealth, рынок мобильной медицины.

Актуальность работы обусловлена развитием одного из направлений мирового здравоохранения – мобильной медицины (mHealth).

Mobile health – это использование в режиме реального времени мобильных устройств для:

- сбора клинических данных о состоянии здоровья пациента;
- снабжения необходимой медицинской информацией врачей и пациентов;
- мониторинга жизненно важных функций пациента;
- прямого предоставления медицинской помощи.

Сегодня mHealth развивается параллельно в двух крупных направлениях, оказывающих влияние друг на друга:

- технологии, устройства, приложения и услуги для лечения и ухода за пациентами;
- системы и устройства, предназначенные для контроля соблюдения здорового образа жизни и фитнеса.

Контроль работы сердца, сна и других жизненно важных процессов, а также электронные карты пациента, телемедицина и удаленный мониторинг – активно внедряются в сферу здравоохранения.

Рынок «умных» медицинских устройств с каждым годом расширяет ассортимент. Появляются новые устройства, производители стараются улучшить их, путем добавления дополнительных функций, это приводит к производству multifunctional устройств.

С помощью таких устройств пользователь имеет возможность производить мониторинг своего физического состояния, а в случае необходимости – отправлять полученные данные врачу [1, 2].

Наиболее популярные медицинские устройства:

- фитнес-трекеры. Это гаджеты, предназначенные для контроля физической активности человека. Основная задача трекеров – подтолкнуть владельцев к активной деятельности и контролировать получаемые нагрузки. В устройство встроены датчики контроля пульса, шагов, затраченных калорий, датчики по контролю уровня стресса, качества сна, скорости

перемещения и пройденного расстояния. Информация с фитнес-трекера может быть передана на смартфон или компьютер, где можно произвести расчеты активности человека и проследить динамику показателей здоровья. Наиболее распространенная форма устройства – браслет;

- часы. Гаджет, как и обычные часы, может показывать время, а также собирать информацию. Для использования полного функционала необходимо синхронизировать со смартфоном. Основное назначение – на экран приходят сообщения о звонках и СМС, поступивших на смартфон, новостях из социальных сетей, напоминания о запланированных событиях, прогноз погоды и т.д. Могут работать как фитнес-трекеры. Некоторые функции работают без подключения к смартфону, например, калькулятор;
- «Умная» одежда – одежда, которая интегрируется с современными информационными технологиями. Она помогает людям с физиологическими отклонениями или заболеваниями. Также «умная» одежда полезна при работе с опасными веществами. Такая одежда позволяет следить за состоянием человека – производить контроль основных жизненных показателей. «Умная» одежда дает возможность удаленно провести анализы и осуществить дистанционное медицинское консультирование. Некоторые модели имеют функцию геолокации;
- обувь. Отслеживать частоту сердцебиения, температуру кожи, уровень кислорода помогают датчики, встроенные в «умные» носки, стельку для обуви или саму обувь. Некоторые модели имеют датчики GPS;
- глюкометр. Устройство, которое помогает непрерывно следить за уровнем глюкозы. Датчик, носимый на теле, производит измерения и с помощью беспроводного соединения передает данные в смартфон;
- весы. Весы, которые взаимодействуют с приложением на смартфоне и отправляют ему результаты измерений. Приложение может построить график изменений веса, вычислить индекс массы тела и другие важные показатели.

Мобильные приложения индустрии mHealth имеют несколько основных направлений [3]:

- профилактика заболеваний и пропаганда здорового образа жизни;
- расчет отдельных диагностических показателей и физиологических параметров;
- осуществление удаленного мониторинга состояния больного, контроль эффективности лечения, приверженности и точности выполнения пациентами лечебных предписаний;
- консультативная (информационная) поддержка врачей и пациентов.

Поскольку сфер применения мобильных приложений в индустрии mHealth большое количество, в данной работе были рассмотрены мобильные приложения в области контроля физиологического состояния.

Мобильные приложения по контролю физиологического состояния.

Одной из самых популярных сфер применения мобильных приложений, наряду с самодиагностикой, является использование различных мобильных приложений для контроля физиологического состояния:

Google Fit и Apple HealthKit позволяют собирать статистику о физической активности человека, пройденных шагах, анализировать информацию и представлять данные в графической форме.

Runtastic дает возможность отслеживать время, путь, калории и другие показатели, а также анализирует кардиотренировки.

Некоторые возможности приложения:

- слежение за пробегом своих кроссовок;
- выбор, сохранение и просмотр своих маршрутов;
- отображение избранных параметров тренировки;
- показ изменения показателей (включая темп, высоту и уклон в виде цветного маршрута).

Waterbalance. Мобильное приложение для слежения за уровнем воды в организме, исходя из его индивидуальных параметров: роста, веса, возраста и образа жизни [4, 5].

Medisafe – это приложение, которое помогает принимать лекарства вовремя по предписанию врача. Дополнительно следит за такими показателями здоровья, как: вес, давление, уровень глюкозы в крови и др.

Health – комплекс приложений, которые заменяет медицинскую карту. В них можно поместить данные о здоровье (аллергии, уровень глюкозы, холестерина и др.). Health синхронизируют данные с другими приложениями в iPhone и формируют обширную картину здоровья.

Lifesum – приложение, которое информирует пользователя об употребляемых им в пищу продуктах. Для того чтобы получить всю информацию о продукте и сохранить ее в приложении, достаточно отсканировать штрих-код на упаковке.

Шагомер. Приложение подсчитывает количество пройденных шагов, одновременно показывая сумму потраченных калорий, пройденное расстояние, время и скорость передвижения пользователя. Поддерживается на Android.

Fitport. Fitport – приложение, собирающее данные вашей физической активности.

MyFitnessPal. Мобильное приложение для мониторинга потребленных калорий и пищевой ценности продуктов. Требуется ручное введение информации о съеденных продуктах.

Motion 24/7 Sleeptracker. Приложение предоставляет подробные ежедневные графики сна и метрик для точного и непрерывного мониторинга частоты дыхания спящего, сердечный ритм, и движения в течение всей ночи. Понятные графики сна визуально различать периоды, когда вы были в фазе быстрого сна, легкого сна, глубокого сна или бодрствования.

Glucose Buddy. Позволяет отслеживать уровень глюкозы в крови, лекарства, питание, отслеживать тенденции в крови сахара, инсулина, веса, артериального давления, гликированного гемоглобина, и др. Приложение дает возможность видеть изменения в уровне сахара в крови и потребление углеводов на почасовой основе. Можно вести журнал пищи с использованием обширной базы данных пищи, автоматически отслеживать шаги прогулок и других видов сердечной деятельности.

Существуют также специальные приложения для контроля отдельных антропометрических и физиологических показателей (индекс массы тела, окружность талии, бедер, грудной клетки и т.д.). Например, приложение iHandy Level, изначально предназначенное для немедицинского применения в качестве инклинометра, как показали исследования, может использоваться и в медицинских целях, и позволяет с достаточной точностью и надежностью измерять подвижность суставов нижних конечностей и уровень их восстановления после травмы.

Популярность приобретают приложения, мотивирующие человека к отказу от вредных привычек, прежде всего, от курения (SmokeFree28, REQ-Mobile, iSmoke и др.). Из значительного числа таких программ эффективность была подтверждена лишь у нескольких, поэтому их широкое применение в повседневной практике требует дальнейшего изучения [5].

Также удобным электронным инструментом являются медицинские калькуляторы, которые позволяют автоматически рассчитывать различные диагностические и физиологические параметры. Созданы универсальные калькуляторы для одновременного расчета нескольких параметров (Medscape, Calculate by QxMD, MediCalc) [4].

Мобильное здравоохранение на сегодняшний день один из значимых проектов в медицине, который позволит поднять на новый уровень доступность и качество врачебной помощи, упростить диагностические процедуры, повысить информированность общества, уменьшить финансовые расходы и улучшить управление системой здравоохранения в целом.

В индустрии мобильной медицины для пользователя первостепенную важность имеют:

- легкий доступ к медицинским услугам;
- возможность самостоятельно следить за своим здоровьем.

От удобства и внешнего вида пользовательского интерфейса мобильного приложения зависит эффективность взаимодействия пользователя с этим приложением и вероятность его длительного использования. Рынок устройств и приложений mHealth – это конкурентная среда, поэтому необходимо найти способ удержать пользователей с самого начала.

Литература

1. Kirichek R., Pirmagomedov R., Glushakov R., Koucheryavy A. Live Substance in Cyberspace – Biodriver System // 18th International Conference on Advanced Communication Technology. – 2016. – P. 274–278.
2. Pirmagomedov R. Ya., Koucheryavy E. A., Glushakov R. I., Kirichek R. V., Koucheryavy A. E. Live Substance in Cyberspace – Biodriver System // *Electrosvyaz*. – 2016. – № 1. – P. 47–52.
3. Никитин П. В., Мурадянц А. А., Шостак Н. А. Мобильное здравоохранение: возможности, проблемы, перспективы // *Клиницист*. – 2015. – № 9(4). – С. 13–21.
4. Bierbrier R., Lo V., Wu R. C. Evaluation of the accuracy of smartphone medical calculation apps // *Journal of Medical Internet Research*. – 2014. – V. 3. – № 16(2). – P. e32.
5. Jacobs M. A., Cobb C. O., Abroms L., Graham A. L. Facebook apps for smoking cessation: a review of content and adherence to evidence-based guidelines // *J. Medical Internet Res.* – 2014. – V. 9. – № 16(2). – P. e205.



Бобрик Ольга Александровна

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4205

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: bobrik_olga94@mail.ru



Шалобаев Евгений Васильевич

Год рождения: 1947

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.т.н., доцент

e-mail: kpd@limtu.ru



Шуклин Дмитрий Анатольевич

Год рождения: 1979

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: do@limtu.ru

УДК 004.9

ОБЗОР СЕРВИСОВ ДЛЯ ВИЗУАЛИЗАЦИИ ДАННЫХ

Бобрик О.А., Шалобаев Е.В., Шуклин Д.А.

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 рассмотрены этапы разработки визуализации данных, дана краткая характеристика каждого этапа, выделены сервисы, которыми можно воспользоваться на каждом этапе, определены их возможности, преимущества и недостатки.

Ключевые слова: визуализация данных, инфографика, анализ данных, сервисы визуализации, информационная графика.

С каждым годом объем и сложность данных увеличиваются гораздо быстрее, чем способности человека их анализировать и обрабатывать. Визуальное представление данных является одним из лучших методов упростить этот процесс. Одной из причин, по которым мы воспринимаем изображения быстрее, чем текст, является то, как мозг обрабатывает информацию. Он считывает данные в виде рисунка сразу, а текст – последовательно [1]. Однако до сих пор, несмотря на широкий спектр применения визуализации, ей довольно часто пренебрегают или же применяют неграмотно, также часто не уделяют достаточное внимание качеству используемых данных. Этим обуславливается актуальность данной работы.

Целью исследования являлось выделение сервисов, применяемых на разных этапах визуализации данных. Исходя из цели, необходимо было решить следующие задачи:

выделить и описать основные этапы визуализации данных, выбрать современные сервисы, используемые на каждом этапе, и проанализировать их.

Разработку визуализации можно разделить на две стадии: работа с данными и работа с дизайном. В первую стадию входят этапы сбора и анализа данных. Во вторую – создание эскиза, прототипа и дизайн макета.

Информационная графика интересна настолько, насколько интересны лежащие в ее основе данные [2]. Данные можно получить следующими способами: от заказчика/компании, почерпнуть из целого ряда сайтов, онлайн-приложений и сервисов или собрать собственноручно через анкеты, опросы и т.п. В первом случае важно проверить полученную информацию и узнать ее первоисточник. Аналитик Нейтан Яу (Nathan Yau) подтверждает важность достоверных источников. Он считает, что верификация данных – «одна из важнейших, если не самая важная, часть графического дизайна» [2]. Для второго варианта получения данных можно воспользоваться [3]:

- сайтами статистических данных: data.gov.ru, tsenomer.ru, wciom.ru;
- сайтами библиотек: library.ru, rba.ru;
- сервисами Яндекса и Google: Wordstat Yandex, Яндекс.Статистика, Google.AdWords;
- источниками в диаграммах и графиках, публикуемых в СМИ, таких, как РБК, NewYorkTimes и др.

Для извлечения данных с сайтов полезно использовать сервисы web scraping или парсинг. Такое программное обеспечение (ПО) помогает собирать информацию от различных провайдеров и сводить ее в одно место для последующего реферирования и анализа. Популярны следующие сервисы: Google Spreadsheet, Import.io, Scrapinghub, Web Scraper. Минусами программ в основном являются отсутствие интеграции по API, возможности обновления данных, ограничение количества страниц в потоке. Также настройка таких сервисов и потраченное время на обучение иногда не соответствует полученной выгоде.

Анализ данных необходим для формирования общей идеи, выделения основных критериев, по которым будет разрабатываться проект [4]. Для сортировки, группировки и обработки информации в зависимости от целей используются разные методы, но можно выделить два варианта используемого ПО: готовые программы и самостоятельно запрограммировать. У обоих способов есть как преимущества, так и недостатки. Готовые решения – это более простой вариант, однако необходимо потратить время на их изучение, и такое ПО часто имеет ограниченный функционал. Среди сервисов, используемых для визуализации можно выделить: Google Data Studio, Tableau, библиотека D3.js, Flowing Data, Microsoft Excel. Часть этих инструментов предназначены для работы с данными и графикой на базовом уровне, другие для более глубокого анализа и визуального исследования [2]. Кроме ограниченного функционала минусом таких программ можно назвать их стоимость, так как многие из них не являются бесплатными. При написании программ самостоятельно используются в основном следующие языки программирования: Python, R, HTML/CSS/JavaScript, PHP, Processing. Нейтан Яу (Nathan Yau) подчеркивает, что навык написания программ позволяет более гибко работать с различными типами данных и лучше к ним адаптироваться [2]. Главная же задача этого этапа, чтобы в результате осталась только необходимая, полезная и актуальная информация, которая будет на макете в виде текстов или графиков.

После тщательной работы с данными приступают к их визуализации. Первой стадией здесь является создание эскиза. Эскиз может быть как условный, так и более подробный, его можно нарисовать и на бумаге, главное понять, как лучше расположить информацию [3]. Далее эскиз переходит в прототип. Цель прототипа определить правильный порядок считывания материала. Павел Лебедев утверждает, что в прототипе необходимо придерживаться следующих принципов [3]:

- самое важное должно занимать больше места и быть самым заметным на макете;

- необходимо расположить весь текст (текст должен быть в том объеме, как будет в финальной версии макета);
- отрисовать все схемы и графики;
- расположить элементы и акценты для правильного порядка считывания;
- оставить все элементы в черно-белых цветах и тоном показать самые яркие объекты;
- не вставлять фотографии и иконки (вместо них можно использовать простые примитивы).

Для разработки прототипов можно воспользоваться следующими сервисами: Infogr.am, Visual.ly, Tableau Public, Piktochart, Easel.ly, Vizualize.me. Они позволяют легко размещать необходимую информацию и элементы, создавать графики на основе таблиц и диаграммы, использовать готовые базы данных, некоторые имеют возможность создания интерактивной инфографики. Однако такие приложения используют шаблоны, которые нет возможности полностью изменить, загружаемые данные не защищаются, некоторые не воспринимают кириллицу или отображают неправильно.

Дизайн макет можно собирать как в известных программах Adobe Photoshop, Adobe Illustrator, Inkscape, так и воспользоваться инструментами, описанными выше. При разработке дизайн макета необходимо уделить внимание модульной сетке, типографике, цвету и иллюстрациям [3]. Для этого также можно воспользоваться определенными приложениями и веб-сайтами.

Для работы с модульной сеткой предлагают следующие сервисы: The Generator, Gridulator, Grid Generator, Grid Designer. С их помощью все элементы на макете могут быть выровнены по визуальным направляющим и расположены в необходимом порядке. Далее следует поработать с текстом: выбрать шрифты, составить набор стилей текста (заголовки, подзаголовки, основной текст, подписи), проверить, чтобы все было читабельно. На этом шаге также существуют вспомогательные сайты: Google Fonts, Type light, Font Flame, Fontstruct и др. Они помогают подобрать шрифт, создать собственный, конвертировать шрифты PostScript, TrueType, OpenType и редактировать их. У многих присутствует большая библиотека шрифтов.

При работе с цветом и выборе цветовой палитры необходимо, чтобы акценты сохранялись такие же, как и на прототипе: самое важное в макете должно привлекать больше внимания [3]. Если сложно придумать собственную цветовую палитру, и отсутствует фирменный стиль для ориентирования, можно воспользоваться несколькими вариантами: взять готовую палитру цветов из другого макета, подобрать цвета в специальном приложении. В первом способе для поиска подойдут такие сайты, как pinterest.com или behance.net. Во втором – сайты и сервисы: Color Hunt, Colorscheme, BrandColors, TinEye и др. В них можно создавать и настраивать цвета, градиенты и схемы, подбирать цветовые сочетания, анализировать палитры изображений.

При размещении фотографий, иллюстраций и иконок на макете можно прибегнуть к онлайн-хранилищам:

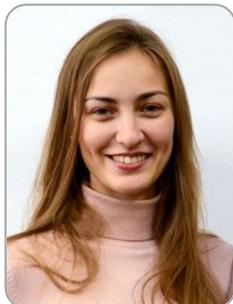
- для иконок: Iconfinder, Icon8, Thenounproject, Iconbird;
- для фотографий: Depositphotos, Shutterstock, Unsplash, Pexels, Gratisography;
- для иллюстраций, векторной графики: Vecteezy, Freepic.

Среди них есть как платные, так и бесплатные сервисы. Платные содержат больше изображений, что позволит сэкономить время на поиск и поможет сделать макет более профессиональным, но бесплатные также хранят достаточное количество материала хорошего качества и постоянно обновляются.

Таким образом, в данной работе были кратко рассмотрены этапы разработки визуализации данных, выделены области, которым стоит уделить внимание, выбраны и проанализированы сервисы, используемые в каждой стадии. Далее будут проанализированы проблемы каждого этапа разработки и выделены методы, помогающие повысить качество визуализаций.

Литература

1. Смиклас М. Инфографика. Коммуникация и влияние при помощи изображений. – СПб.: Питер, 2014. – 152 с.
2. Яу Н. Искусство визуализации в бизнесе. Как представить сложную информацию простыми образами / Пер. с англ. С. Кировой. – М.: Манн, Иванов и Фербер, 2013. – 352 с.
3. Лебедев П. Курс для дизайнеров: визуализация данных и дизайн инфографики [Электронный ресурс]. – Режим доступа: <http://kurs.skillsup.ru/infografika>, своб.
4. Лаптев В.В. Информационный дизайн и визуализация данных // Электронное научное издание «Дизайн. Теория и практика». – 2014. – № 15. – С. 32–45.



Букирёва Людмила Александровна

Год рождения: 1993

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: bukirevala@gmail.com

УДК 004

ИССЛЕДОВАНИЕ ХАРАКТЕРИСТИК CMS НА ПЛАТФОРМЕ NODE.JS

Букирёва Л.А.

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 системы управления контентом на платформе Node.js. Выбран ряд CMS и критерии для их исследования. Проанализированы характеристики каждой из исследуемых CMS, а также проведено сравнение функциональных характеристик исследуемых CMS.

Ключевые слова: система управления контентом, CMS на платформе Node.js, характеристики CMS, Ghost, KeystoneJS, Apostrophe, Pencilblue.

Рост серверного JavaScript за последние годы привел к появлению различных CMS, построенных на платформе Node.js из-за ее быстрой скорости. Эти CMS имеют открытый исходный код, благодаря чему разработчики могут анализировать его и улучшать CMS, внося свои изменения с согласия авторов.

На данный момент существует большое количество исследований, посвященных сравнительному анализу систем управления контентом, где рассмотрены классификации систем управления контентом, выделены критерии для сравнения CMS, а также разработаны рекомендации для применения той или иной системы для определенного вида веб-ресурса (интернет-магазин, блог и т.д.). Однако в этих работах объектами исследования являются CMS только на платформе PHP [1]. CMS на платформе Node.js мало изучены в силу своей новизны.

Вышесказанное подтверждает недостаточную изученность в научных исследованиях проблемы выбора CMS на платформе Node.js для разработки сайтов и определяет актуальность работы.

Согласно рейтингу и рекомендациям различных интернет-сообществ можно выделить наиболее популярные CMS для дальнейшего исследования: Ghost, KeystoneJS, Apostrophe и Pencilblue.

Проанализировав критерии для сравнения систем управления контентом [2, 3], а также определив функциональные свойства CMS, способствующие комфортной и продуктивной разработке того или иного интернет-ресурса, были выделены следующие критерии для сравнения CMS на платформе Node.js: популярность, активность разработки, пользовательские типы записей, дополнительные поля, плагины, на основе MVC, документация для разработки темы, поддержка сообщества, документация.

1. Популярность. Критерий популярности можно считать одним из самых показательных в оценке качества CMS, так как он основывается на отзывах пользователей, которые тестировали систему в реальных проектах. В исследовании была использована оценка популярности от 0 до 10, взятая из ресурса «Awesome JS», где предоставляется актуальная информация о частоте использования различных CMS на Node.js.

2. Активность разработки. Данный критерий характеризует степень активности сообщества разработчиков в создании CMS. По этому показателю можно выявить, насколько активно развивается проект. В исследовании была использована оценка активности разработки от 0 до 10, взятая из ресурса «Awesome JS», где предоставляется актуальная информация об интенсивности коммитов в репозиториях проектов на GitHub.
3. Пользовательские типы записей. Как правило, в CMS существует возможность создания записи, формат которой предусмотрен в базовом функционале системы. Однако иногда возникает необходимость разместить контент в формате, который не предусмотрен CMS «из коробки». Тогда и появляется необходимость создания кастомизированной записи, имеющей нестандартное представление. Примером предоставления подобного функционала могут стать пользовательские типы записей (Custom Post Types) в CMS Wordpress [4]. Для оценки данного показателя будет применяться система оценки, где 0 – отсутствие возможности добавления пользовательского типа записи, 1 – наличие, 0,5 – функциональность CMS отличается концептуально.
4. Дополнительные поля. Во многих системах управления контентом предусмотрены стандартные поля, с помощью которых можно публиковать записи. Однако не всегда этих полей достаточно, поскольку иногда требуются дополнительные поля для более расширенной информации. В этой связи очень полезно и удобно, когда в CMS есть возможность добавить пользовательские поля, которые могут представлять собой текстовую область, стандартную форму для ввода текста, выпадающие опции, календарь, форму выбора цвета или загрузки изображений. Например, в CMS Wordpress для этих целей используется плагин Advanced Custom Fields, а в CMS Joomla! есть уже встроенная возможность использования пользовательских полей (Custom fields). Для оценки данного показателя будет применяться система оценки, где 0 – отсутствие возможности добавления дополнительного поля, 1 – наличие, 0,5 – функциональность CMS отличается концептуально.
5. Плагины. Поддержка различных плагинов для дополнительного функционала системы делает CMS более гибкой и пригодной для большего ряда задач. Во многих популярных CMS, реализованных на языке PHP, имеется такая возможность. Таким образом, система с поддержкой плагинов будет в преимуществе. Для оценки данного показателя будет применяться следующая система оценки: 0 – отсутствие поддержки плагинов, 1 – наличие небольшого количества плагинов, 2 – наибольшее количество плагинов.
6. На основе MVC. Шаблон проектирования MVC предполагает разделение данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: Модель, Представление и Контроллер – таким образом, что модификация каждого компонента может осуществляться независимо [5]. CMS, основанная на MVC, сделает разработку более стандартной и отлаженной. Для оценки данного показателя будет применяться двоичная система оценки, где 0 – отсутствие шаблона MVC в основе, а 1 – наличие.
7. Документация для разработки темы. Как правило, в системе управления контентом присутствует набор готовых дизайнерских решений для оформления сайта. Однако не всегда бывает достаточно базовых предложений. Для этого разработчикам предоставляется возможность создания собственной темы и соответствующая документация для реализации идей на данной платформе. Оценка данного параметра будет производиться в соответствии со следующими правилами: 0 – отсутствие документации для разработки темы, 0,5 – неполная документация/отсутствие документации компенсируется, 1 – наличие документации.
8. Поддержка сообщества. При возникновении той или иной проблемы с CMS, разработчик может задать вопрос людям, которые тоже использовали данную систему и, возможно, сталкивались с подобной проблемой, и знают, как ее решить. Таким образом, важно выяснить, есть ли у CMS поддержка сообщества. Для оценки данного показателя будет

применяться следующая система оценки: 0 – нет поддержки сообщества/слабая поддержка, 1 – есть активная поддержка сообщества.

9. Документация. Наличие информативной документации – один из важных критериев при выборе CMS, поскольку отсутствие качественной документации к продукту может негативно повлиять как на скорость, так и на качество разработки. Оценка данного параметра будет производиться в соответствии со следующими правилами: 0 – отсутствие документации/плохая документация, 1 – наличие качественной документации.

Результаты оценки каждой CMS по вышеперечисленным критериям представлены в таблице.

Таблица. Анализ характеристик CMS

	Ghost	KeystoneJS	Apostrophe	Pencilblue
Популярность	9,7	8,9	4,2	4,0
Активность разработки	9,6	8,7	9,4	3,5
Пользовательские типы записей	0	1	0,5	1
Дополнительные поля	0	0,5	0,5	1
Наличие плагинов	0	0	1	2
На основе MVC	1	1	1	1
Документация для разработки темы	1	0,5	1	0,5
Поддержка сообщества	1	1	1	0
Документация	1	1	1	1
ИТОГО	23,3	22,6	19,6	14

Из результатов анализа CMS по 9 отобраным критериям можно сделать следующие выводы. Наиболее популярной CMS среди разработчиков является CMS Ghost. Также эта система выделилась самым высоким показателем активности разработки. Следующими по популярности соответственно располагаются KeystoneJS, Apostrophe и Pencilblue. Несмотря на то, что KeystoneJS оказалась второй по популярности, активность ее разработки ниже, чем у CMS Apostrophe. Наименьшая активность разработки наблюдается у Pencilblue. Все четыре CMS одинаково обладают следующими характеристиками: основаны на шаблоне проектирования MVC, имеют качественную документацию на официальном сайте. Однако только у Ghost и Apostrophe есть надлежащая документация для разработки тем. Полноценная настройка дополнительных полей, а также большой набор плагинов есть в Pencilblue, однако только у данной системы самая слабая поддержка сообщества среди конкурентов. Возможность настройки пользовательских типов записей есть у систем KeystoneJS и Pencilblue. Несмотря на самую высокую популярность, Ghost имеет наибольшее количество «нулей» в оценке критериев исследуемых систем, тогда как следующая за ней KeystoneJS и Pencilblue имеют только одну нулевую оценку, а у Apostrophe «нули» и вовсе отсутствуют.

Исходя из итоговой оценки CMS, состоящей из суммы всех оценок, можно заметить, что Ghost и KeystoneJS лидируют, а Pencilblue имеет наименьшую оценку. Однако если в данном сравнении CMS не учитывать популярность и активность разработки, которые не отражают функциональных характеристик CMS, итоговая оценка окажется примерно одинаковой для всех исследуемых CMS.

Большинство из исследуемых CMS имеют аспекты управления контентом, отличающиеся от традиционных систем, такие как WordPress, Joomla или Drupal. Ghost специализируется на блогах. KeystoneJS и Pencilblue являются полнофункциональными CMS. Apostrophe поднимает редактирование интерфейса на следующий уровень.

Ghost не имеет дополнительных плагинов, возможности добавления кастомизированных типов записей и полей, поэтому идеально подойдет для создания и ведения блога, где эти функции не являются необходимыми.

KeystoneJS подойдет для разработчиков сайтов или людей, которые собираются заказать создание и настройку системы у профессиональных разработчиков. Многие идеи и характеристики данной CMS представляются интересными и свежими, а автоматически генерируемый интерфейс администратора упрощает создание сложных веб-сайтов и приложений.

Apostrophe имеет все необходимые характеристики для создания достойного сайта. Она подойдет для разработчиков сайтов и клиентов, лидирующих на рынке какой-либо услуги, которые готовы выделить из своего бюджета средства на разработку легко редактируемой CMS. Однако нужно быть уверенным в том, что WYSIWYG, лежащее в основе Apostrophe, будет использоваться клиентом правильно.

Pencilblue выглядит и работает очень похоже на WordPress и в некотором смысле данная CMS лучше организована. Система имеет процветающую систему плагинов, что может оказаться очень полезным качеством для разработки. Pencilblue имеет широкую целевую аудиторию, поскольку это полнофункциональная платформа, и ее можно использовать для разных типов сайтов.

Литература

1. Захарова М.В., Горяйнова Е.В. Система управления контентом сайта (на примере сайта газеты «Зори» Северского района Краснодарского края) // Молодой ученый. – 2016. – № 20(124). – С. 311–313.
2. Дуйсегалиева А.Д. Сравнительный анализ систем управления контентом для создания интернет-магазина: дис. ... магистра: 09.04.02. – СПб.: Университет ИТМО, 2016. – 73 с.
3. Семенова Т.Ю. Сравнительный анализ систем управления контентом: дис. ... магистра: 09.04.02: защищена 10.06.2016. – СПб.: Университет ИТМО, 2016. – 76 с.
4. Conțu С.А., Popovici Е.С., Fratu О., Verceanu M.G. Security issues in most popular content management systems // International Conference on Communications. – 2016. – P. 277–280.
5. MVC для веб: проще некуда [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/181772/>, своб.



Ванина Наталья Валерьевна

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4206

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: nata.vanina.v@mail.ru

УДК 004.925.5

**ВОЗМОЖНОСТИ НЕЙРОННЫХ СЕТЕЙ В РАСПОЗНАВАНИИ ЦВЕТА
ИЗОБРАЖЕНИЙ**

Ванина Н.В.

Научный руководитель – к.ф.-м.н. Соловьев И.А.

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

В работе рассмотрены особенности нейросетевых технологий, в частности – сверточных нейронных сетей, проанализированы возможности данных инструментов в процессе решения тривиальной задачи распознавания цвета или цветосочетаний изображения, приведен пример работы фильтра, позволяющего разделить изображение на отдельно исследуемые сектора для дальнейшего сравнения и получения карты цветосочетаний.

Ключевые слова: нейросетевые технологии, сверточные нейронные сети, распознавание цвета, распознавание цветосочетаний, юзабилити интерфейса.

Одной из самых тривиальных, но не менее важных задач компьютерных технологий, является распознавание цвета и цветосочетаний. В контексте возрастающей популярности информационных сетей и актуализации юзабилити интерфейса, появляется потребность в грамотном подборе цветовой палитры для различных пользователей, в том числе и с отклонением цветовосприятия (дальтонизмом).

Хотя в настоящее время уже существуют технологии распознавания элементов по цвету (OpenCV3) [1], данный способ требует перевода изображения RGB в цветное пространство HSV, схему которого можно наблюдать на рис. 1, а, что достаточно ресурсозатратно и не продуктивно при обработке большого массива данных. Для решения данной проблемы (в ходе исследовательской работы) требовалось изучить и проанализировать нейросетевые технологии, найти способы минимизировать затраты ресурсов и облегчить процесс определения цвета или цветосочетаний изображения путем обучения нейросетевой модели.

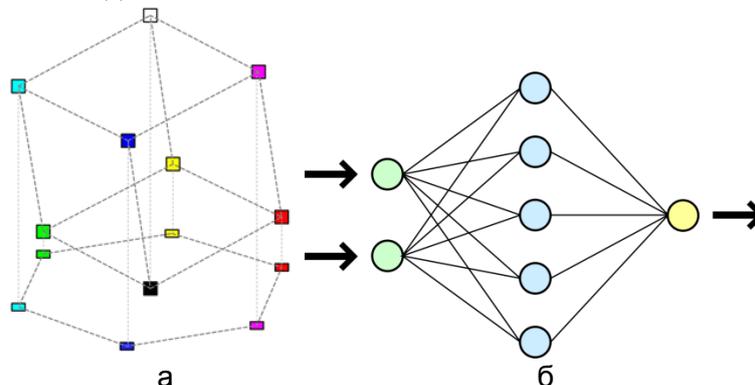


Рис. 1. Переход из цветовой модели RGB в HSV (а); схематическое представление нейронной сети (б)

Нейронные сети уже давно являются неотъемлемой частью развития IT-технологий и постепенно внедряются в различные сферы человеческой деятельности. Особое внимание исследователи уделяют распознаванию изображений и обработке их в соответствии с заданными признаками, что позволяет не только находить искомые объекты, но и классифицировать их. Нейронная сеть имеет возможность вычленить потенциально опасные цветовые сочетания интерфейсов для пользователей с нарушением цветовосприятия, базируясь на базу данных уже проанализированных примеров, что поможет устранить проблему на момент разработки дизайна проекта.

Причина популярности и повышенное внимание к данному способу машинного обучения состоит в том, что данная модель (искусственная нейронная сеть) берет за основу современные представления о способе обработки информации в человеческом мозге, способного приспосабливать и переорганизовывать свои структурные компоненты (нейроны) для решения сложных, конкретных и плохо формализованных задач, таких как распознавание элементов по заданным свойствам и др.

С. Хайкин в своей книге «Нейронные сети. Полный курс» дает такое определение данного понятия: «Нейронная сеть – это громадный распределенный параллельный процессор, состоящий из элементарных единиц обработки информации, накапливающих экспериментальные знания и предоставляющих их для последующей обработки» [2].

Данные технологии позволяют разбивать процесс обработки объекта на несколько параллельных задач, с последующим обобщением полученных данных, что наглядно представлено на рис. 1, б.

Сверточные нейронные сети (СНС) – это одна из разновидностей нейронных сетей, которая нацелена на эффективное распознавания изображения и используется для решения как тривиальных задач (в данном случае она фокусируется на определении цветосочетания), так и сложных проблем классификации [3].



```

08 02 22 97 38 15 00 40 00 75 04 05 07 76 52 12 50 77 31 08
49 49 99 40 17 81 18 57 40 87 17 40 98 43 69 48 04 56 42 00
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 45
52 70 95 23 04 60 11 42 49 24 48 56 01 32 56 71 37 02 34 91
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 43 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 43 72
21 34 23 09 75 00 76 44 20 45 35 14 00 41 33 97 34 31 33 95
78 17 53 28 22 75 31 47 15 94 03 80 04 42 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 94 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 46
88 36 48 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 49
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 49 34 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 97 05 94
01 70 84 71 83 51 94 49 16 92 33 48 41 43 52 01 89 19 47 48

```

Рис. 2. Сравнение человеческого и компьютерного зрения

Как показано на рис. 2, компьютерное зрение значительно отличается от человеческого и представляет собой набор из цифры, представленный в виде матрицы.

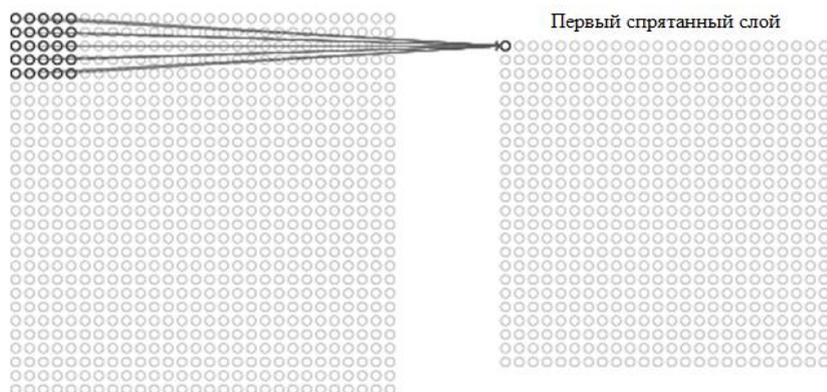


Рис. 3. Визуальное представление фильтра 5×5

Сверточная нейронная сеть позволяет обрабатывать данный набор числовых данных и относить изображение к определенному классу. Так, при распознавании цвета объекта на изображении, исследуется сначала отдельные его детали, расположенные на одинаковых участках, и только потом полученные данные обобщаются и позволяют нейронной сети отнести данное изображение к одной из заданных классификаций. На рис. 3 показан фильтр (матрица весов или матрица параметров), и размеры этого фильтра – $5 \times 5 \times 3$ [4].

Поскольку фильтр производит свертку, т.е. передвигается по вводимому изображению, он умножает значения фильтра на исходные значения пикселей изображения (поэлементное умножение). Все эти умножения суммируются (всего 75 умножений). И в итоге получается одно число [5]. На рис. 4 представлена карта цветов, полученная в ходе и обработки и анализа изображения. Сочетания полученных цветов в клетках анализируются, классифицируются как правильное или потенциально опасное сочетание, заносится в базу данных нейронной сети и используется как образец дальнейших распознаваний цветов.

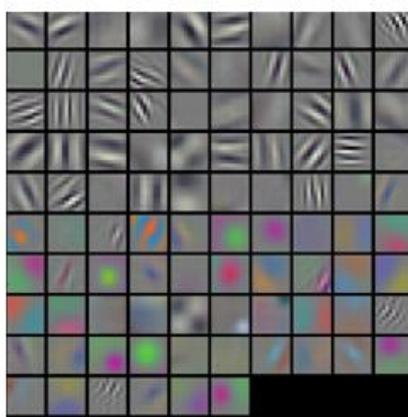


Рис. 4. Визуализация цветового фильтра

В данной работе представлено краткое описание нейросетевых технологий, и их возможностей в процессе выполнения тривиальной задачи распознавания цвета или цветосочетаний для дальнейшей классификации как опасное или безопасное при показе изображения пользователям с нарушением цветовосприятия. На основе изученных данных и уже ранее проведенных исследований, был составлен и визуализирован фильтр для распознавания цвета и описан принцип работы сверточной нейронной сети.

В ходе дальнейшего исследования планируется создание и полноценная апробация сверточной нейронной сети для распознавания нежелательных цветосочетания интерфейса для дальтоников, база данных которой будет состоять из реальных или искусственно созданных примеров с ошибочным подбором цветовой палитры.

Литература

1. Abrial P. Color detection for vision machine defect inspection on electronic devices // Electronic Manufacturing Technology Symposium. – 2010. – P. 180–192.
2. Хайкин С. Нейронные сети: полный курс. – 2-е изд. – М.: Вильямс, 2006. – 1104 с.
3. Deshpande A.A Beginner's Guide to Understanding Convolutional Neural Networks [Электронный ресурс]. – Режим доступа: <https://adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/> (дата обращения: 02.03.2017).
4. LeCun Y., Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W. and Jackel L.D. Backpropagation Applied to Handwritten Zip Code Recognition // Neural Computation1. – 1989. – V. 4. – P. 541–551.
5. Nielsen M. Neural Networks and Deep Learning [Электронный ресурс]. – Режим доступа: <http://neuralnetworksanddeeplearning.com/> (дата обращения: 02.03.2017).

**Варганов Дмитрий Анатольевич**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207cНаправление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: varganovda@gmail.com

УДК 004.4'22

ОБЗОР ПОСТПРОЦЕССОРОВ CSS И ИХ ЗАДАЧ**Варганов Д.А.****Научный руководитель – к.т.н., доцент Сокуренок Ю.А.**

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

В работе рассмотрены существующие инструменты по пост-обработке стилевых файлов с целью упрощения задач верстальщиков и фронтенд-разработчиков. Описаны основные возможности данных обработчиков. Указаны основные недостатки препроцессоров и их использования.

Ключевые слова: CSS, препроцессоры, постпроцессоры, PostCSS, LESS, SASS.

Развитие технологий все больше сказывается на жизнедеятельности людей. Так, например, раньше требовалось совершать множество механических действий для получения результата, например, покупка молока требовала посещения магазина, а запись к врачу занимала большое количество времени в очередях. С ростом прогресса в сети Интернет и веб-технологий, в частности, многие задачи сводятся к простому взаимодействию с компьютером или мобильным устройством [1–5].

Так, согласно исследованию, в 2017 году количество пользователей глобальной сети составило 3,58 млрд человек, при этом общее население планеты составляет 7,6 млрд. Согласно статистике, количество уникальных пользователей мобильных устройств составляет 5 млрд человек и продолжает ежедневно расти. Таким образом, можно утверждать, что сфера веб-разработки будет оставаться актуальной продолжительное время.

Однако прогресс технологий разработки сайтов также не стоит на месте. Ежегодно в области появляется множество новых инструментов для ускорения и упрощения процесса написания программного кода. Так, например, появились пре- и постпроцессоры. Ежедневно появляются порядка 5–10 новых из них.

После продолжительной борьбы с языком CSS и его проблемами, сообщество веб-разработчиков долго искало возможности для улучшения приемов работы с разрабатываемыми проектами. Появившиеся в начале 2000-х годов инструменты для структурирования кода стилевых файлов, стали быстро захватывать рынок разработки, это были препроцессоры. Препроцессор – это надстройка над языком программирования, добавляющая дополнительные возможности в начальный язык без вмешательства в стандартные механизмы. Первым из инструментов стал SASS. Он появился в 2006 году и занял позицию основного инструмента для улучшения структуры и читабельности кода на языке CSS. Первоначально SASS был написан на языке Ruby, но в связи с его низкой производительностью, в 2009 году появился новый инструмент под названием LESS, предназначенный для совместного использования с языком JavaScript, на котором был написан, а далее был переписан и первопроходец на многих языках программирования. В настоящее время самыми популярными препроцессорами являются LESS, SASS(SCSS), Stylus, но чаще всего используются первые два. Препроцессоры решают задачи по

уменьшению каскада стилей, упрощению кода, его читабельности, использовать универсальные механизмы для повторяющихся или ветвящихся стилей для описания множества схожих объектов и т.д.

Постпроцессоры – это специальные обработчики, которые на вход получают код на обыкновенном языке CSS и модифицируют его согласно выбранным настройкам, на выходе получается код на обычном CSS.

В процессе использования препроцессоров возникают следующие проблемы:

1. применяется специальный синтаксис, который не разбирают браузеры;
2. невозможность интеграции кода в CSS и JavaScript файлы без предварительной обработки;
3. снижение скорости отображения стилей;
4. малое число проектов, в которых применяются все возможности инструментов.

Самыми известными постпроцессорами на данный момент являются `autoprefixer` и `postcss`, первый в настоящее время стал плагином для второго.

Определить рамки задач постпроцессора практически невозможно, потому что для `postcss` на данный момент существует более 2000 плагинов и число продолжает расти. Например, одними из задач пост-обработчиков являются: минификация кода, интеграция инлайн-изображений в кодировке `base64`, линтеры, подсветка связей HTML и CSS-кода, создание спрайтовых изображений, объединение файлов, группировка селекторов по выражениям, автоматическое документирование, автоматическое удаление заранее заданных участков кода и неиспользуемого кода, автоматический поиск шрифтов, замена символов на необходимые и т.д.

Зачастую для использования постпроцессоров требуются программы сборщики: `grunt`, `gulp` или `webpack`; для некоторых существуют библиотеки с графическим интерфейсом или платформа `node`. Сами плагины и пост-обработчики следует выбирать индивидуально, требования к их использованию можно найти на ресурсах, через которые они распространяются.

Таким образом, следует вывод, что для каждого рода задач следует выбирать инструменты, которые им соответствуют. Например, не стоит нагружать препроцессоры задачами по установке префиксов через миксины.

Литература

1. `Autoprefixer`. GitHub [Электронный ресурс]. – Режим доступа: <https://github.com/postcss/autoprefixer> (дата обращения: 13.12.2017).
2. `PostCSS`. GitHub [Электронный ресурс]. – Режим доступа: <https://github.com/postcss/postcss> (дата обращения: 15.12.2017).
3. Хабрахабр. CSS-препроцессоры против постпроцессоров [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/235929/> (дата обращения: 25.11.2017).
4. WebforMyself. Постпроцессоры CSS для начинающих: советы и ресурсы [Электронный ресурс]. – Режим доступа: <https://webformyself.com/postprocessor-css-dlya-nachinayushhix-sovety-i-resursy/> (дата обращения: 25.11.2017).
5. Мясников М., Городилов Ю.Н., Надводцкая В.В. Оптимизация и повышение производительности CSS-кода // Материалы XVI международной научно-технической конференции. – 2015. – С. 129–132.

**Водова Варвара Александровна**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4106Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: vodova94@mail.ru

**Готская Ирина Борисовна**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, д.педагог.н., профессор

e-mail: iringot@mail.ru

УДК 555.32

**ИССЛЕДОВАНИЕ ТЕХНОЛОГИЙ ПЕРСОНАЛИЗАЦИИ КОНТЕНТА
И ПОЛЬЗОВАТЕЛЬСКОГО ОПЫТА В МОБИЛЬНЫХ ПРИЛОЖЕНИЯХ****Водова В.А., Готская 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».

Потенциальные клиенты различных видов бизнеса тонут в информации и предложениях в сети Интернет, в связи с этим Веб в эпоху Web 3.0 приобретает элементы интеллектуальности. Сегодня в мобильных и вебприложениях активно используются технологии персонализации контента и пользовательского опыта, они призваны сократить объем данных и количество предоставляемых предложений и функций, чтобы устремить целевую аудиторию в глубину воронки конверсии, сформированной на основе предполагаемых индивидуальных потребностей аудитории.

Ключевые слова: мобильные приложения, персонализация контента, персонализация пользовательского опыта, технологии персонализации.

На сегодняшний день технологии персонализации используются в большинстве веб и мобильных приложениях. Sitecore и Vanson Bourne провели опрос в 2016 году, который показал, что персонализация – это наиболее востребованный у пользователей мобильных гаджетов элемент дизайна [1]. Персонализация сайта – это автоматическая подстройка информации на сайте в соответствии с интересами каждого пользователя либо сегмента пользователей. Персонализация идентифицирует конкретных пользователей, либо устанавливает их характеристики на основе прошлых действий (обращений в службу поддержки сайта, просмотра определенных страниц, поиска определенной информации на сайте, покупки в интернет-магазине и др.) [2]. На Западе считается устоявшимся следующее определение персонализации: «процесс создания мобильного приложения таким образом, чтобы оно удовлетворяло нуждам определенной аудитории» [3].

Есть гипотеза, что потенциальные клиенты превращаются в лиды в зависимости от того, насколько правильно персонализирован сайт или пользовательский опыт. Но какие типы персонализации контента маркетологи задействуют в работе чаще всего? Обратимся к

данным опроса компании Evergage, опубликованном в 2015 году [4], результаты которого размещены на рис. 1, 2.

Вопрос: Какие типы персонализированных сообщений или опыта вы используете? (Отметьте все варианты, которые подходят).

What types of personalized web messages/experiences do you use? (Select all that apply.)

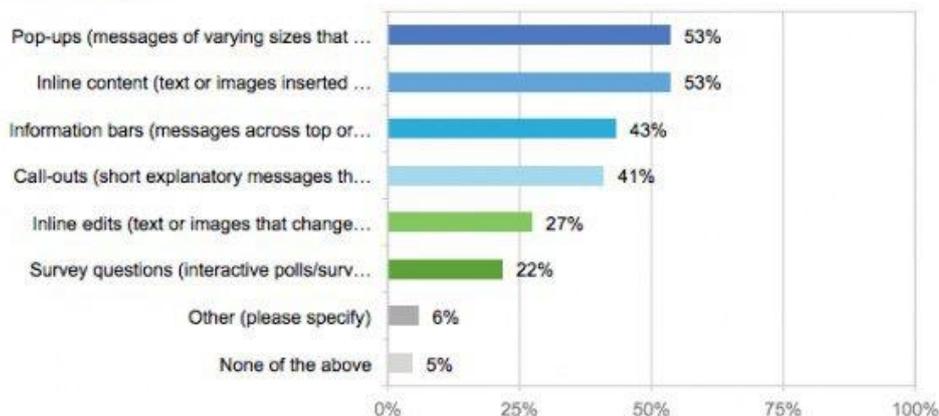


Рис. 1. Типы наиболее популярных персонализированных сообщений или опыта

1. всплывающие окна – 53%;
2. inline-контент – 53%;
3. информационные блоки – 43%;
4. подсказки – 41%;
5. встроенное редактирование – 27%;
6. опросы – 22%;
7. другое – 6%;
8. ничего из перечисленного – 5%.

Персонализация работает с серьезными объемами данных, таким образом, существует большое разнообразие вариаций при выборе сегментирования аудитории. Большинство выбирают тип просматриваемого контента, географическое положение и количество времени, проведенное на сайте в общей сложности.

Which of the following methods do you use for segmenting visitors or users in order to do real-time personalization? (Select all that apply.)

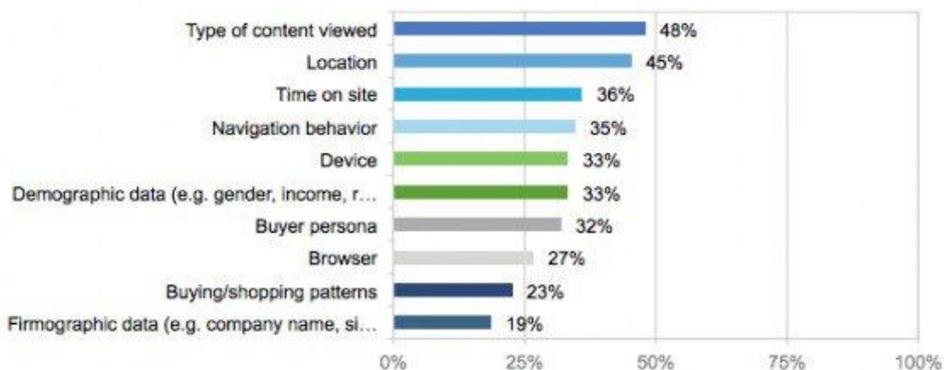


Рис. 2. Наиболее популярные методы сегментирования посетителей с целью персонализации

Вопрос: Какой из нижеперечисленных методов вы используете для сегментирования посетителей с целью персонализации в режиме реального времени? (Отметьте все варианты, которые подходят):

- тип просматриваемого контента – 48%;
- местоположение – 45%;

- время, проведенное на сайте – 36%;
- поведение посетителя на сайте – 35%;
- устройство – 33%;
- демографические данные – 33%;
- персона покупателя (buyerpersona) – 32%;
- браузер – 27%;
- покупательское поведение – 23%;
- данные о компании – 19%.

Таким образом, многие маркетологи уделяют большую роль персонализации контента и менее скромную роль индивидуальному пользовательскому опыту. Чтобы персонализация работала наиболее эффективно, показа всплывающих окон во время прихода/ухода на/с сайта недостаточно, необходимо придерживаться более оригинального подхода и учитывать различные раздражающие факторы. Сценарии персонализации контента формируются в зависимости от вида приложения, установленных целей и используемых технологий персонализации.

Существуют различные классификации технологий персонализации. Персонализация включает в себя три аспекта: взаимодействие, содержание и опыт [5]. Исследователи классифицируют системы персонализации по следующим критериям: по типам концептуальных уровней, на которых осуществляется персонализация; по методам сбора входных данных; по подходам к фильтрации информации; по методам персонализации; по входным данным модели пользователя; по использованным данным; классификация по принципу «внутри/вне» мобильных приложений [6].

Анна Левинсон приводит следующий пример классификации [7]:

1. персонализация внутри приложения: в приложении персонализация основывается на таких функциях, как всплывающие диалоговые окна, базирующиеся на размещении или предпочтениях, поисковые рекомендации, скидки и специальные предложения;
2. персонализация вне приложения: вне приложения, персонализация осуществляется благодаря push-уведомлениям. Такие опции как дефолтные поисковые запросы, автозаполнение форм на основе расположения, скидки и специальные предложения, предположения, основанные на предыдущих использованиях, создаются с целью улучшения персонализации и удобства использования приложения. Но при неправильной реализации, эффект может быть прямо противоположным, поэтому необходима качественная аналитика.

ИТ-рынок также адаптируется под тенденцию использования персонализации и в ответ на растущий спрос предлагает системы по созданию адаптивного контента. Системы адаптивного контента делают из простой веб-страницы «мультилендинги» – демонстрируемый пользователю на экране контент (заголовки и содержание текста) автоматически персонализируется под поисковый запрос, по которому клиент попал на сайт после проведения соответствующих настроек.

Сегодня одним из ключевых инструментов digital-маркетинга являются платформы DMP. Data Management Platform (DMP) – мультифункциональная система, позволяющая сохранять и систематизировать данные первого и второго порядка (first-party, second-party data) и дополнять их данными третьего порядка (third-party data).

В процессе исследования было выявлено общее представление о положении технологий персонализации контента и пользовательского опыта в мобильных приложениях и методах их реализации; обобщены отечественные и зарубежные исследования, посвященные теме технологий персонализации контента и пользовательского опыта в мобильных приложениях; выделены научные задачи, требующие дальнейшего исследования, в том числе и задача освещения темы современных методов и типов персонализации контента и пользовательского опыта в мобильных приложениях, а также современных систем и платформ, предназначенных для персонализации пользовательского опыта.

Литература

1. Emarketer. InternetUsersExpectMorePersonalizedMobileApps, Sites [Электронный ресурс]. – Режим доступа: <https://www.emarketer.com/Article/Internet-Users-Expect-More-Personalized-Mobile-Apps-Sites/1013962/>, своб.
2. Яковлев А.А., Ткачев В.Г. Раскрутка сайтов: основы, секреты, трюки. – СПб.: БХВ-Петербург, 2015. – 356 с.
3. Персонализация мобильных приложений [Электронный ресурс]. – Режим доступа: <http://uxgu.ru/mobile-app-personalization/>, своб.
4. Почему персонализация контента это еще не веб-персонализация? [Электронный ресурс]. – Режим доступа: <https://lpgenerator.ru/blog/2016/03/19/pochemu-personalizaciya-kontenta-eto-eshe-ne-veb-personalizaciya/>, своб.
5. Три стратегии персонализации, которые гарантируют успех вашего приложения [Электронный ресурс]. – Режим доступа: <http://apps4all.ru/post/05-23-17-3-strategii-personalizatsii-kotorye-garantiruyut-uspeh-vashego-prilozheniya-216/>, своб.
6. Царев А.Г. Управление контентом веб-сайта на основе персонализации данных: дис. канд. техн. наук: 05.13.01. – М., 2011. – С. 18–22.
7. Hannah Levenson. MobileAppPersonalization: HowToDoItRight [Электронный ресурс]. – Режим доступа: https://usabilitygeek.com/mobile-app-personalization-how-to/?utm_source=UsabilityGeek+Newsletter&utm_campaign=b3c3886f75-RSS_EMAIL_CAMPAIGN&utm_medium=email&utm_term=0_107a1a5dab-b3c3886f75-94935577/, своб.

**Волобуев Никита Геннадиевич**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4208Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: mr.bunny194@gmail.com

**Готская Ирина Борисовна**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, д.педагог.н., профессор

e-mail: iringot@mail.ru

УДК 004.4**СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРЕДСТАВЛЕНИЙ НА ПРИМЕРЕ ПЛАТФОРМ
ASP.NET И DJANGO****Волобуев Н.Г.****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 сравнение двух наиболее популярных веб-фреймворков ASP.NET и Django, а именно, возможности, которые они предоставляют для построения пользовательского интерфейса. Разобрана реализация паттернов MVC и MTV. Изучены возможности скаффолдинга, применяемые в данных фреймворках для автоматического генерирования представления на основе объектно-ориентированного подхода.

Ключевые слова: ASP.NET, Django, MVC, MTV, View, Template, скаффолдинг, пользовательский интерфейс.

В настоящее время глобальная сеть Internet стала использоваться в очень многих областях нашей жизни. Все больше людей взаимодействуют с Интернетом: посещают разнообразные сайты и используют приложения на смартфонах. В связи с этим появилось множество технологий, предоставляющих возможность создания и разработки web-интерфейсов различной сложности. Учитывая актуальность данной темы, в настоящей работе ставилась следующая цель – исследовать способы построения пользовательского интерфейса с применением одного из самых популярных шаблонов проектирование в веб-разработке MVC (MTV) на примере технологий Django и ASP.NET.

Платформа ASP.NET MVC является ничем иным как фреймворком для написания сайтов и веб-приложений, в основе которой лежит реализация паттерна MVC. Данная терминология, а именно MVC, начала применяться приблизительно в конце 1970-х годов, когда она стала использоваться в проекте Smalltalk. В своих истоках она задумывалась как один из способов построения интерфейса приложений. Некоторые начальные функции первичных версий паттерна MVC были взяты из ранее существующих понятий Smalltalk, например, экран или инструменты. Однако более глобальные термины все еще используются в приложениях, в особенности они очень хорошо описывают веб-приложения.

Работа приложения, написанного с применением MVC, схожа с естественным циклом взаимодействия пользователя с обновлением представлений, в данном случае представление можно считать, не умеющим сохранять свое состояние. Зачастую это также хорошо сопоставляется с HTTP-запросами и ответами, которые находятся в самом центре веб-приложений.

Помимо этого, MVC пытается разделить термины: модель доменов и логику, отвечающую за управление контроллерами, которые зачастую отделены от интерфейса пользователя. В веб-приложениях это значит, что HTML пишется самостоятельно (в отдельной части от всего приложения), это позволяет сделать поддержку и тестируемость в разы проще и легче.

В стандартном случае, опираясь на паттерн MVC, можно сказать, что структура приложения будет разбита минимум на три части:

1. модель – механизм, хранящий в себе или представляющий данные, которые запрашивает пользователь. Модели бывают разных типов, например, простая модель – это модель, предназначенная для представления данных пользователю, которые отдает контроллер. Это может быть и доменная модель – модель, хранящая в себе доменные данные и операции, а также, возможно, обладающая функционалом преобразования данных;
2. представление – инструмент, предназначенный для того, чтобы построить пользовательский интерфейс, используя данные из моделей;
3. контроллер – инструментарий, необходимый для обработки входящих запросов, выполняющий различные операции, необходимые для функционирования моделей, и решающий, какое из представлений использовать в данный момент.

Модель является ничем иным как отдельной частью, в которой работает каждое приложение. Модель хранит всю информацию о предметной области и умеет предоставлять ее в дозированном формате, она также в ответственности за сохранение и целостность данных, например, модель должна гарантировать, что данные будут внесены в базу данных (БД) и будут корректны на протяжении всего времени существования приложения. Модель не должна заниматься интерфейсом и обработкой запросов, этот функционал должны реализовывать контроллеры. В то же самое время представление должно включать в себя логику, которая будет необходима при отображении частей модели пользователю. Представление ни в коем случае не должно напрямую общаться с моделью, для этой задачи существует контроллер, представляющий из себя соединительный механизм между моделью и представлением. Работа происходит по следующему принципу: в тот момент, когда приходит запрос от пользователя, контроллер обрабатывает его, при этом выбирая необходимое представление, которое нужно показать пользователю, а иногда в данный момент могут выполняться и дополнительные шаги, манипулирующие моделью.

Абсолютно каждый фрагмент, включенный в архитектуру MVC фреймворков, должен быть автономным. Это означает, что должно присутствовать понятие, называемое разделением обязанностей. Логика, отвечающая за манипуляцию с данными в моделях, должна присутствовать только в моделях и нигде больше. Код, отвечающий за обработку запросов, хранится только в специально написанных для этого контроллерах. При сохранении такой архитектуры на протяжении всего пути разработки приложения, разработчикам будет намного проще поддерживать и расширять его.

В asp.net контроллер представляет собой не что иное, как обычный класс, написанный на одном из языков семейства .net, зачастую он является производным от System.Web.Mvc.Controller класс. Каждый метод, который обладает модификатором доступа уровня public и одновременно является производным от Controller, имеет возможность называться методом действия, напрямую связанным с системой роутинга в asp.net, и настраивается с помощью прописывания URL-шаблонов. Каждый раз, когда на сервер отправляется запрос на какой-то URL-адрес, с ним связывается метод, который выполняет определенные действия в контроллере, тем самым воздействует на доменную модель и

определяет, какое представление нужно отобразить в текущем случае. На рисунке можно более детально рассмотреть этот процесс.

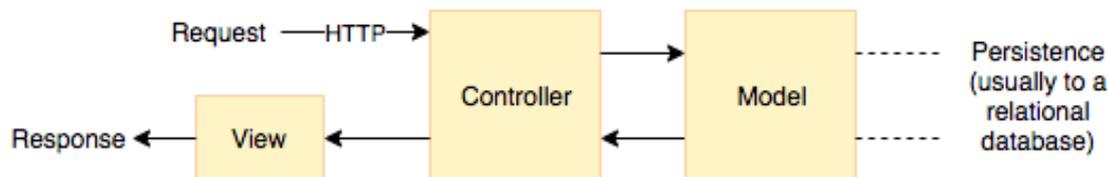


Рисунок. Процесс взаимодействия компонентов

ASP.NET фреймворк обладает возможностью выбора движка, который будет отвечать за работу представления. В первых версиях фреймворка использовался базовый движок, который обрабатывал страницы aspx с помощью усовершенствованной версии разметки на базе web forms. С выходом в свет новой версии фреймворка под цифрой 3 также был представлен новый движок, носивший название Razor, который в дальнейшем был усовершенствован в последующих версиях, и было принято решение использовать совершенно другой синтаксис. В Visual Studio изначально встроен пакет, обеспечивающий поддержку IntelliSense для обоих движков, что в значительной мере упрощает работу. В ASP.NET не существует никаких ограничений на реализацию доменной модели. Разработчики вольны спроектировать модель с помощью обычных классов и хранить свои данные в любой удобной для них БД.

Основными инструментами для управления шаблонами в Django используются теги, фильтры и переменные. Все переменные, которые передаются в шаблоны в момент рендеринга, замещаются своим значением в текущем контексте. Синтаксически переменная передается в двойных фигурных скобках, в следующем виде: `{{ data }}`. Фильтры – это механизм, необходимый для преобразования переменных, они могут быть представлены как в виде переменных, так и в виде качества параметра тега. Синтаксически это можно представить в следующем виде: имя переменной | название фильтра: «параметр». Во время отрисовки шаблона все теги замещаются функциями, написанными на питоне, которые ассоциируются с конкретным шаблоном.

Например: `{% url my article slug = article.slug %}`

Существуют специализированные теги, а именно: `include`, `block` и `extend`. Основным является `include`, который подставляет требуемый шаблон для отрисовки в текущий момент. Одним из главных достоинств Django является наследование. Оно реализовано с помощью тегов `block` и `extend`. Любой шаблон способен расширить собой поведение родительского шаблона. В любом месте внутри шаблона можно вставить блочный тег, за исключением случаев, что тег начинается перед циклом и заканчивается внутри него. Также каждому блоку дается имя.

Например:

```

{% block content %}
тело блока
{% endblock %}
  
```

С помощью тега `extend` есть возможность обозначить, какой шаблон необходимо расширить. Во время расширения шаблона становится доступна возможность переопределить необходимые блоки, которые присутствуют в родительском шаблоне. На выходе получаем довольно функциональный инструмент, благодаря которому практически исключается необходимость дублирования частей шаблонов.

Опираясь на вышесказанное и на работы российских [1–3] и зарубежных [4–6] ученых о технологии построения представления с помощью шаблонов Django, можно сделать следующие выводы:

- множество шаблонных тегов упрощают построение логических конструкций;

- слабая связанность компонентов: любой участник команды способен работать независимо друг от друга, что позволяет ему вносить изменения в проект, не вмешиваясь в чужой код;
- разработка сайтов становится гораздо проще благодаря механизму наследования шаблонов. Такое решение является более верным, чем дополнение кода с помощью тега `{% include %}`. Смысл наследования состоит в написании родительского и дочернего шаблона. Родитель содержит в себе общие разделы, и в нем описываются переопределяемые в унаследованных шаблонах блоки. Переопределение становится возможным при добавлении в дочерние страницы тега `{% extends «index.html» %}`, где `index.html` – главная страница.

Технология ASP.NET MVC от компании Microsoft также обладает рядом преимуществ при построении View, а именно:

- встраиваемые методы HTML-помощника предоставляют соответствующие стандартам выходные данные, вместо того чтобы плодить огромные участки HTML, которым сложно управлять;
- MVC рекомендует выработать простой, элегантный стиль разметки с помощью CSS;
- возможность использования строго типизированных представлений;
- мощный движок Razor.

Литература

1. Мэтиз Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. – СПб.: Питер, 2017. – 496 с.
2. Васильев П.А. Web-программирование на языке Python. Фреймворки Django, Flask // Наука, техника, образование. – 2016. – С. 38–39.
3. Шаблоны Django. Наследование [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/23132/> (дата обращения: 20.02.2018).
4. Downey A. Think Python: How to Think Like a Computer Scientist. – Green Tea Press, Needham, Massachusetts, 2008. – 189 с.
5. Django vs Flask vs Pyramid: Choosing a Python Web Framework [Электронный ресурс]. – Режим доступа: <http://www.airpair.com/python/posts/django-flask-pyramid> (дата обращения: 19.02.2018).
6. Django vs Pyramid vs Flask [Электронный ресурс]. – Режим доступа: <http://www.geo-viz.com/blog/django-vs-pyramid-vs-flask> (дата обращения: 19.02.2018).

**Волошин Олег Владимирович**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4206Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: o.voloshin@protonmail.com

УДК 004.422.83

**РАЗРАБОТКА УТИЛИТЫ СТРУКТУРИЗАЦИИ ЭВРИСТИЧЕСКИХ МЕТОДОВ
В ПРОИЗВОДСТВЕННЫХ ПРОЦЕССАХ****Волошин О.В.****Научный руководитель – к.т.н., доцент Шалобаев Е.В.**

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

Проведен анализ характеристик подходов к структуризации эвристических методов. На основе выявленных закономерностей сформулирована и предложена модель данных для утилиты моделирования систем эвристических методов, обеспечивающую наглядность данных и гибкость самой системы.

Ключевые слова: оптимизация, эвристические методы, промышленный дизайн, визуализация.

Оптимизация методов проектирования и производственного дизайна является одним из путей решения проблемы неэффективного использования ресурсов и потери времени. Такая оптимизация касается и подходов к созданию продукции, в частности и к первоначальной ступени планирования жизненного цикла продукта и задач дальнейшего проектирования и производства.

Это влияет и на подход к разработке продукта с применением эвристических методов, базирующегося на опыте инженера, интуиции и предположений и часто не имеющего под собой строгой алгоритмизированной базы. Эвристические методы являются инструментом генерирования идей, изначально разработанным с целью помочь инженерам в поиске вариативных инженерных решений [1]. Работа Лоры Мёрфи описывает эвристические методы как одних из самых эффективных подходов в рамках процесса генерации идей с точки зрения степени концентрации инженера непосредственно на функциональности компонента [2], над которым ведется работа.

Под эвристическими методами подразумеваются приемы и правила для определения основных направлений работы и генерации идей, основанные на процессе инженерного творчества и научного опыта. Преимуществами подхода, основанного на таких приемах, является возможность вольного проектирования, способного привести к конечной цели при неполных исходных данных или отсутствии строго сформулированного плана по достижению искомого решения. Важно упомянуть, что эвристические методы не гарантируют достижения благоприятного результата, но оказывают значительную помощь в определении возможных решений [3].

Для повышения вероятности благоприятного исхода при использовании данного подхода предпринимались разработки различных конвенций и методологий структуризации эвристических методов [4]. Существующие подходы подвергались анализу, в результате которого предлагались новые методы структуризации, зачастую основанные на комбинации свойств уже существующих схем.

В работе «Методики структуризации эвристических методов в производственных процессах», также выполненной в рамках данной исследовательской работы, был произведен анализ различных задокументированных подходов к структуризации эвристических систем и предложен новый – с использованием следующих требований для формирования системы эвристических методов [4]:

- использовать понятные и краткие предложения в тексте инструкции эвристического метода;
- составлять для каждой идеи не более одного эвристического метода;
- избегать составления противоречащих друг другу эвристических методов;
- составлять эвристические методы только при условии должного обоснования для их применения;
- брать в расчет разброс в уровне профессиональной подготовки, опытности и принадлежности к разным сферам деятельности пользователя;
- при возможности использовать дополнительную контекстную информацию;
- подкреплять текстовые инструкции эвристических методов визуальной составляющей, содержащей необходимые технические детали (например, схемы взаимодействия компонентов системы или физические параметры модуля);
- балансировать наличие визуальной составляющей с содержанием эвристической инструкции;
- допускать незначительный уровень избыточности эвристических систем для обеспечения их стабильности;
- группировать эвристические методы согласно контексту применения.

Данные инструкции опираются на общие правила, зачастую используемые во многих существующих методиках, и были использованы для создания модели данных для системы управления эвристическими методами. Для этого необходимо определить набор характеристик, которые будут сочетать в себе наиболее стабильные существующие решения и соблюдение списка требований выше.

Для выявления базовой части свойств эвристических методов для модели данных был рассмотрен анализ различных подходов к структуризации эвристических методов в работе [5]. Данная работа демонстрирует, что в каждом из них, так или иначе, присутствуют следующие характеристики:

- название/краткое описание/тезис;
- описание/инструкции;
- пример;
- взаимосвязи;
- доп. параметры (пример: цель, источник, основные вопрос и т.д.).

Таким образом, первые четыре атрибута лишь незначительно варьируются от системы к системе и формируют собой неизменное «ядро» атрибутов, остающихся постоянными от системы к системе. Остальные параметры эвристической системы часто являются специфичными для контекста их составления или уточняющими, например: «Assessment questions: How can fulfilment of the guideline be assessed?» (перевод: «Оценочные вопросы: Как будет оцениваться выполнение инструкции?») [5, С. 7, табл. 4].

Модель данных для утилиты была основана на предоставлении гибкости в настройке параметров системы, при этом она содержала в себе встроенные ключевые свойства – аналоги неизменных параметров из рассмотренного анализа. Факт того, что вариативность в структуризации эвристических систем часто определяется лишь контекстом, свидетельствует о том, что такая структура системы упростит ее конфигурацию и использование, так как она позволяет сосредоточиться лишь на специфичных для контекста свойствах, не теряя времени на установление базовых параметров, которые с большой вероятностью будут дублироваться среди большинства систем. Кроме того, важно предоставить данные о взаимодействии всех эвристических предписаний системы. Для этого будут использованы два вида взаимосвязей.

Главной сущностью модели данных является эвристический метод. Для всех эвристических методов в рамках одной коллекции (систематизированной совокупности эвристических предписаний) свойства являются идентичными и определены параметрами самой коллекции. При создании коллекции ей будет присвоен набор свойств по умолчанию:

- название;
- описание/инструкции;
- контекст применения;
- ответственная единица.

Допускается также и конфигурация дополнительных свойств для всех элементов системы, если проект требует специфичного планирования, например, необходимо разбить предписания по масштабу их внедрения. Значения свойств (не затрагивает атрибуты название и описание) предлагается реализовать на основе системы тегов. Это означает, что одинаковые значения свойства будут объединены в одну сущность, что, в свою очередь, позволит эффективнее категоризировать элементы системы.

Помимо перечисленных атрибутов, каждая единица системы потенциально может иметь два вида взаимосвязей с другими звеньями: реализация и воздействие. Связь типа реализация представляет собой иерархическое взаимоотношение между звеньями систем и устанавливается между ними по принципу от общего к частному. Например, в случае «для реализации А нужно выполнить В и С», элемент А будет иметь две исходящих связи типа реализация к элементам В и С. Связь типа воздействие, в отличие от реализации, обладает не только направлением (источник – цель или родитель – приемник), но и булевым значением соответствующему положительному или отрицательному показателю влияния одного звена системы на другое.

Такое устройство модели позволяет организовать эвристические методы в форме направленной иерархической структуры, позволяющей наглядно анализировать систему эвристических методов как изображено на рисунке.

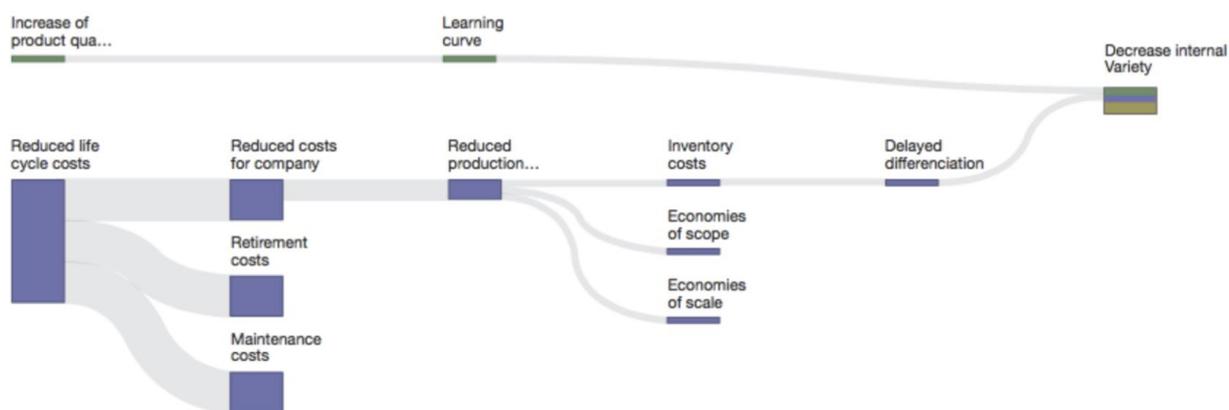


Рисунок. Визуализация системы эвристических методов

В результате пользователь получает визуализированную модель процесса разработки продукции. Модель структурирована как с точки зрения иерархии звеньев (позволяет легко визуально отделять ветки разработки по определенным направлениям или контекстам), так и относительно характера их взаимодействия. Таким образом, проведя сравнительный анализ некоторых подходов по решению вопроса организации систем эвристических методов, была сформулирована модель данных, которая станет основой для утилиты, предназначенной для хранения и структуризации и визуализации эвристических методов [6].

Литература

1. Daly S.R., Yilmaz S., Christian J.L., Seifert C.M. and Gonzalez R. Design Heuristics in Engineering Concept Generation // J. Eng. Educ. – 2014. – V. 101. – № 4. – P. 601–629.

2. Laura M., Daly S.R., Yilmaz S. and Seifert C.M. Supporting Novice Engineers in Idea Generation using Design Heuristics [Электронный ресурс]. – Режим доступа: <https://works.bepress.com/seda-yilmaz/26/download/>, своб.
3. Yilmaz S., Seifert C.M. and Gonzalez R. Cognitive heuristics in design: Instructional strategies to increase creativity in idea generation // *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*. – 2010. – V. 24. – № 3. – P. 335–355.
4. Волошин О.В. Методики структуризации эвристических методов в производственных процессах // *Альманах научных работ молодых ученых XLVI научной и учебно-методической конференции Университета ИТМО*. – 2017. – Т. 4 – С. 47–49.
5. Sarnes J. and Hermann K. Heuristic Guidelines In Ecodesign // *Proceedings of the 20th International Conference on Engineering Design*. – 2015. – V. 1. – P. 171–180.
6. Bonvoisin J., Mathieux F., Domingo L., Brissaud D. Design for energy efficiency: proposition of a guidelines-based tool. In *International Design // Conference-DESIGN*. – 2010. – 10 p.

**Воробьев Алексей Игоревич**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления «Академия ЛИМТУ», кафедра компьютерного проектирования и дизайна, студент группы № S4108

Направление подготовки: 09.04.02 – Информационные системы и технологии

e-mail: zenti.asteris@gmail.com

**Государев Илья Борисович**

Университет ИТМО, факультет методов и техники управления «Академия ЛИМТУ», кафедра компьютерного проектирования и дизайна, к.педагог.н., доцент

e-mail: goss@corp.ifmo.ru

УДК 004.652

**АНАЛИЗ ПРОБЛЕМ МИГРАЦИИ БАЗ ДАННЫХ ВЕБ-ПРИЛОЖЕНИЙ
ОТ РЕЛЯЦИОННОЙ МОДЕЛИ****Воробьев А.И., 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 модели баз данных, используемых в современных веб-приложениях, определены случаи, когда может возникнуть вопрос о миграции базы данных от реляционной модели, а также рассмотрены основные проблемы, которые могут возникнуть при выполнении процесса миграции.

Ключевые слова: базы данных, SQL, NoSQL, веб-приложения, миграция баз данных.

Многие современные веб-приложения принимают, хранят, обрабатывают и выводят информацию, получаемую из различных источников. Для выполнения этой задачи используются базы данных (БД). Существует несколько различных моделей БД, и наиболее популярной на сегодняшний день является реляционная модель, которая обеспечивает хранение данных в виде таблиц и функционирует по принципам алгебраических множеств, используя язык структурированных запросов SQL [1].

Широкое распространение реляционных БД и их систем управления обусловлено тем, что реляционные СУБД обеспечивают хранение и оптимизацию данных, что позволяет сэкономить разработчикам приложений значительную часть времени и ресурсов. Кроме того, реляционная модель данных подходит для представления большинства данных, используемых при работе веб-приложений.

Однако при развитии проектов, объемы и сложность данных, с которыми им приходится работать, могут увеличиться, что способно привести к тому, что реляционная модель перестанет удовлетворять всем требованиям веб-приложения, и возникнет необходимость рассмотрения вопроса о смене БД.

Основные причины для миграции БД от реляционной модели:

1. проект использует большие данные, что добавляет требование линейной масштабируемости и высокой скорости обмена данными;

2. необходимо обеспечить независимость данных от расположения, что подразумевает чтение и запись данных независимо от того, где физически происходят эти процессы;
3. требуется обеспечить непрерывную доступность веб-приложения, например, с использованием механизма децентрализации и рассредоточения данных по нескольким дата-центрам;
4. требуется обеспечить большое разнообразие данных, включая структурированные и неструктурированные данные.

Среди существующих моделей БД наиболее оптимальным выбором будут являться БД NoSQL. NoSQL означает «не только SQL» и является технологией, спроектированной для предоставления приложениям возможности преодоления ограничений реляционных БД на масштабируемость, производительность и распределение данных [2]. Однако в отдельных случаях оптимальным может являться переход и к другим моделям.

При реализации такого процесса, как миграция БД веб-приложения от одной модели к другой, возникают определенные проблемы. В первую очередь необходимо выбрать наиболее оптимальную БД с нужной архитектурой из числа существующих решений. После этого необходимо определить способ миграции:

1. создавать новые приложения и модули с использованием новой модели поверх существующей реляционной системы, что позволит избежать необходимости переписывать существующие компоненты и мигрировать существующие данные;
2. частично модифицировать систему, мигрируя отдельные компоненты на новую модель, в то время как основная система останется реляционной, что позволит решить конкретные проблемы системы, например, проблему масштабируемости;
3. полностью мигрировать систему на новую модель БД [3].

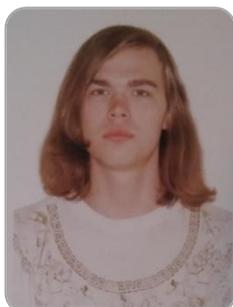
Кроме того, при миграции может возникнуть проблема переноса скриптов, триггеров и прочих особенностей существующей БД.

Весь описанный процесс ставит еще одну важную проблему, заключающуюся в определении стоимости миграции. Зачастую бывает крайне сложно оценить ресурсы, которые понадобятся на осуществления миграции БД, а также оценить преимущества, которые будут получены после ее выполнения. Из-за этого возникает вопрос целесообразности переноса БД, поскольку сложно оценить, окупятся ли в будущем ресурсы, потраченные на миграцию, и, если окупятся, то за какие сроки.

На основе проведенного анализа можно сделать вывод о том, что проблемы при миграции БД веб-приложений от реляционной модели являются актуальными, и необходима методология, которая бы помогала в выборе целевой БД, а также позволила бы помочь ответить на вопрос целесообразности самой миграции.

Литература

1. Structured Query Language. Techopedia [Электронный ресурс]. – Режим доступа: <https://www.techopedia.com/definition/1245/structured-query-language-sql> (дата обращения: 12.01.2018).
2. NoSQL Databases. Datastax [Электронный ресурс]. – Режим доступа: <https://www.datastax.com/nosql-databases> (дата обращения: 12.01.2018).
3. Why Migrate from MySQL to Cassandra. Datastax [Электронный ресурс]. – Режим доступа: <https://www.datastax.com/wp-content/uploads/2012/08/WP-DataStax-MySQLtoCassandra.pdf> (дата обращения: 13.01.2018).



Гаврилов Дмитрий Владиславович

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4106

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: snarepluha@yandex.ru

УДК 7.092

ИССЛЕДОВАНИЕ ТЕХНОЛОГИИ ПЕЧАТИ СИМВОЛИКИ КИБЕРСПОРТИВНЫХ ОРГАНИЗАЦИЙ

Гаврилов Д.В.

Научный руководитель – к.педагог.н. Шуклин Д.А.

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

В работе рассмотрены виды современной печати и применение определенных технологий, которые используются при нанесении элементов фирменного стиля киберспортивных организаций на различную продукцию. Определена киберспортивная продукция, на которую будет наноситься фирменный стиль и предпочтительные методы печати с учетом ее особенностей. Также показывается актуальность работы и современное развитие данной отрасли.

Ключевые слова: киберспорт, фирменный стиль, печать, фирменная продукция, компьютерная графика, визуализация.

Киберспорт на сегодняшний день активно набирает популярность и, являясь частью спорта, большими темпами занимает свою нишу, вследствие чего появляются все новые и новые киберспортивные организации. Одним из способов продвижения имени организации, является создание и распространение продукции, имеющей собственный фирменный стиль. Как и обычные организации в спорте, киберспортивные организации имеют схожую, типичную для спорта продукцию, такую как форма, бренд-волл, ролл-ап, различные канцелярские принадлежности. Но учитывая некоторую специфичность данного занятия, появляется особенная продукция, которая может быть причастна только к киберспорту [1].



а

б

в

Рисунок. Игровое кресло (а); игровая компьютерная мышь (б); коврик для компьютерной мышки (в)

Игровая продукция:

- компьютерные корпуса;
- различные игровые устройства (компьютерные мышки, клавиатуры, наушники) (рисунок, а, б);
- игровая мебель (компьютерные кресла, столы);
- игровые аксессуары (коврики для мышки) (рисунок, в).

Следует учитывать, что при нанесении фирменного стиля на такую оригинальную продукцию, используются способы печати, отличающиеся от стандартной печати. Необходимо учитывать форму предмета, материал, на котором будет держаться краска, и как часто окрашенная поверхность будет контактировать со сторонними предметами и подвергаться физическим воздействиям.

Одним из самых распространенных способов печати является – термотрансфер. Суть данной технологии печати основывается на том, что перед переносом изображения на нужную поверхность, оно печатается на определенном носителе. Затем носитель контактирует с предметом, на который необходимо нанести печать и под воздействием высокой температуры, происходит перенос изображения с носителя на предмет. Данный способ требует специального оборудования, но он достаточно прост, поэтому пользуется большой популярностью. Изображение получается достаточно стойким и, несмотря на присутствие высоких температур во время процесса, цветовая палитра практически не изменяется [2], т.е. потери цвета практически не существенны. Применить его можно практически к любой продукции, но приоритетно его использовать на тканях и жаростойких материалах (при производстве металлических корпусов и фирменной одежды).

Для печати на пластиковых корпусах игровых устройств в приоритет ставится метод ультрафиолетовой печати. Он также обладает очень хорошей стойкостью при контакте с другими телами и большим спектром применений, но, следовательно, более затратным, нежели термотрансфер. Кроме пластика, данный способ отлично подходит для печати на коже, которой обшиваются игровые кресла. И очень удобен при печати на не ровных поверхностях, таковыми могут являться пластиковые части от наушников и компьютерных мышей.

Принцип такой технологии следующий. Перед самим процессом поверхность, на которую будет наноситься изображение, обезжиривается и отчищается от пыли. Затем на нее наносятся ультрафиолетовые чернила, которые застывают на поверхности при воздействии ультрафиолетовых лучей [3]. Из-за своей хорошей износостойкости данный способ применяется предпочтительно к продукции, подвергающейся постоянному физическому взаимодействию. Такой продукцией являются: компьютерные кресла, мышки, клавиатуры и коврики.

Существует еще один способ нанесения изображения на продукцию, но в данной сфере он применяется при создании какого-либо эксклюзивного устройства или предмета. Это – аэрография. При этом краситель является порошковым или жидким, и наносится на поверхность при помощи воздуха, находящегося под высоким давлением. После этого поверхность лакируется для того, чтобы избежать потери цвета и износа изображения [4]. Это наиболее затратный способ нанесения изображения, и его очень проблематично довести до автоматизма, поэтому зачастую он делается в ручную, и является неким произведением искусства, поэтому продукция является эксклюзивной. Наиболее часто этот способ используется при покраске компьютерных корпусов, компьютерных мышек и клавиатур.

Здесь приведены основные виды печати изображений, которые могут применяться при создании фирменной продукции киберспортивных организаций, но способов печати существует намного больше, а способы их применения ограничиваются только творческим мышлением.

Литература

1. Петрова В.Д., Коршунов А.А. Маркетинговый продукт и понятие маркетинг-микса в индустрии спорта [Электронный ресурс]. – Режим доступа: <https://iyusov.livejournal.com/1861.html>, своб.
2. Чивичина О.А. Бережной Н.А. Основные способы печати по ткани [Электронный ресурс]. – Режим доступа: compuart.ru/article/25250, своб.
3. Куприянов Р.А. Что такое УФ-печать? [Электронный ресурс]. – Режим доступа: <http://miosag.ru/uf-pechat.html>, своб.
4. Русич В.С. Что такое аэрография, виды аэрографии [Электронный ресурс]. – Режим доступа: <http://fb.ru/article/194338/что-такое-аерографиya-tehnika-i-stili-aerografii>, своб.



Гарифуллин Аскар Айратович

Год рождения: 1993

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: askar.payment@gmail.com



Сокуренок Юрий Андреевич

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», к.т.н., доцент

e-mail: kpd@limtu.ru

УДК 555.32

**МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ДОБЫЧИ ПОЛЕЗНЫХ ИСКОПАЕМЫХ
НА ВЕБ-ПЛАТФОРМЕ**

Гарифуллин А.А., Сокуренок Ю.А.

Научный руководитель – к.т.н., доцент Сокуренок Ю.А.

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

В работе рассмотрены основные вопросы, связанные с темой моделирования в веб-среде: платформа Electron.js, компоненты математической модели, а также применение микросервисной архитектуры в веб.

Ключевые слова: Electron.js, микросервисы, моделирование, JavaScript.

Современные исследования и разработки в области моделирования процессов нефте- и газодобычи ограничиваются, по большому счету, настольными приложениями, реализованными на Enterprise-языках уровня Java и C#. Тенденции развития интернет-технологий, однако, неизбежно вносят свою лепту в сообщество программистов: популярность языка JavaScript (ECMAScript) растет с каждым годом, на его основе разрабатываются новые технологии и библиотеки, развивается веб-программирование в целом. В то же время в 70-е годы прошлого столетия был введен собирательный термин «интеллектуальное месторождение» [1], обозначающий интенсивное применение цифровых технологий в процессах добычи нефти и газа.

Тем не менее, необходимо учитывать, что вычислительная сложность приложений, основанных на изложенной выше концепции, довольно высока, что ограничивает применение JavaScript ввиду однопоточного характера языка. Действительно, в этом плане JavaScript пока не может тягаться с языками программирования вроде Python, Java, C# и др. (даже несмотря на попытки имитирования многопоточности за счет объектов «WebWorker»), однако использование микросервисной архитектуры позволяет решить эту проблему, что подтверждается исследованиями таких российских ученых, как Ю.С. Артамонов и С.В. Востокин [2], которые говорят о том, что «декомпозиция научных приложений, связанных с вычислениями и обработкой данных на микросервисы, имеет потенциальные

преимущества, известные в области промышленных систем: увеличение гибкости системы, снижение трудоемкости ее сопровождения, упрощение масштабирования и другие».

Сервисная архитектура представляет собой некий набор инкапсулированных сервисов (рисунок), каждый из которых обладает независимым состоянием, базой данных и имеет возможность коммуницировать с другими сервисами посредством протокола передачи данных (например, HTTP) посредством технологии REST (representational state transfer – передача состояния представления). Согласно исследованию [3], микросервисы в последнее время набирают популярность и постепенно становятся новым архитектурным феноменом в области разработки программного обеспечения.

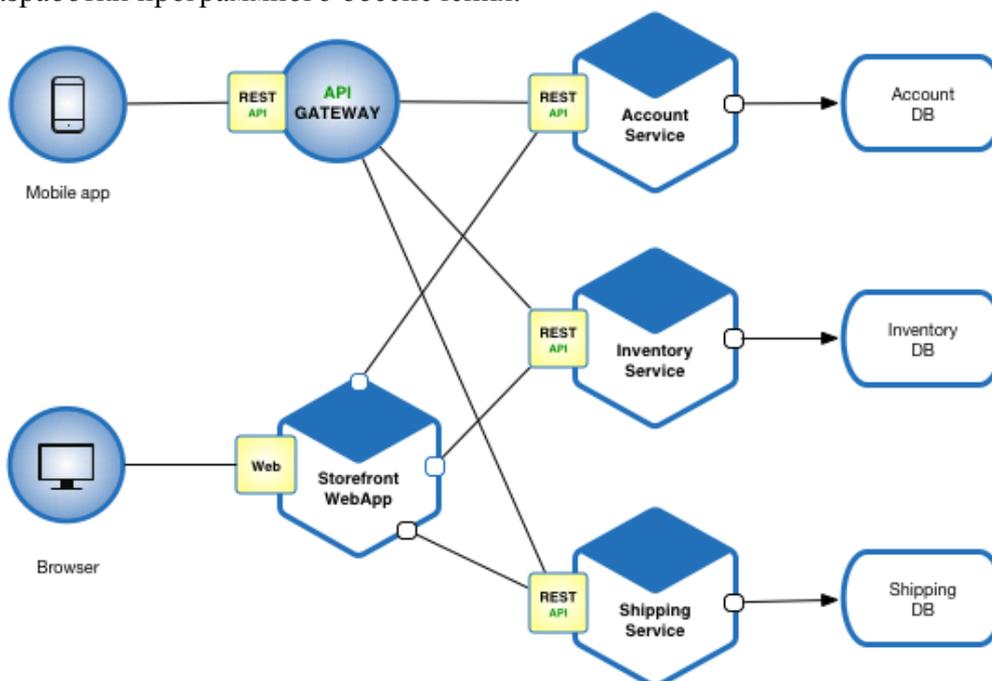


Рисунок. Схематичное представление сервисной архитектуры приложения

Как уже было сказано ранее, развитие веб-сообщества послужило толчком для создания новых технологий и библиотек, одной из которых является Electron.js. Electron.js является инструментом для создания кроссплатформенных приложений при помощи стандартных средств веб-разработки, таких как JavaScript, HTML и CSS. По сути, Electron.js – веб-платформа, позволяющая имплементировать недоступные ранее десктопные приложения средствами стандартного стека веб-технологий. В рамках данного исследования рассмотрена возможность использования Electron.js для построения приложения, моделирующего процессы технологий нефте- и газодобычи.

Непосредственно моделирование основывается на использовании математической (интегрированной) модели, включающей в себя, как минимум:

1. численные методы вычисления: системы линейных алгебраических уравнений;
2. теорию оптимального управления, в частности, стохастические дифференциальные уравнения и дифференциальные уравнения в частных производных, а также метод максимального правдоподобия.

Кроме того, графической составляющей моделирования является интерактивный визуальный редактор, позволяющий добавлять, редактировать и удалять элементы (тройник, разветвитель, труба и др.). Как правило, для представления подобного рода интерактивных компонентов используется SVG-графика, которая получила широкое распространение в вебе.

Подводя итог, можно сказать, что современное состояние веб-технологий в общем, и экосистемы JavaScript в частности, позволяет разрабатывать гибкие, кроссплатформенные приложения, позволяющие решать задачи моделирования разного уровня.

Литература

1. Обзор систем моделирования и инженерных расчетов, применяемых в нефтегазовой отрасли [Электронный ресурс]. – Режим доступа: <http://sstu.syzran.ru/epa/docs/ITiOvNGO/4.2.pdf> (дата обращения: 18.12.2017).
2. Артамонов Ю.С., Востокин С.В. Разработка распределенных приложений сбора и анализа данных на базе микросервисной архитектуры // Изв. Самарского научного центра РАН. – 2016. – № 4-4. – С. 688–693.
3. Asik T., Selcuk Y.E. Policy enforcement upon software based on microservice architecture // Software Engineering Research, Management and Applications. – 2017. – P. 283–287.

**Гарматина Ирина Александровна**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4105Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: ira_garm@mail.ru@mail.ru

**Готская Ирина Борисовна**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, д.педагог.н., профессор

e-mail: iringot@mail.ru

**Шуклин Дмитрий Анатольевич**

Год рождения: 1979

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: do@limtu.ru

УДК 004**ВЛИЯНИЕ АНИМАЦИИ ИНТЕРФЕЙСОВ НА UX ВЕБ-САЙТОВ****Гарматина И.А.****Научный руководитель – д.педагог.н., профессор Готская И.Б.****Консультант – к.педагог.н., доцент Шуклин Д.А.**

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

В работе рассмотрена проблема влияния анимации интерфейсов на пользовательский опыт (UX), результаты зарубежных и отечественных исследований данной проблемы, роль анимации как функционального элемента интерфейса веб-сайта. Получен набор функций анимации интерфейса, качеств функциональной анимации, принципов и приемов улучшения пользовательского опыта за счет применения анимации в веб-дизайне.

Ключевые слова: анимация, пользовательский опыт, UX, веб-сайт, графический интерфейс, когнитивная нагрузка.

В последнее время все больше дизайнеров используют анимацию как функциональный элемент. Анимация используется уже не просто для зрелищности, сейчас это один из важнейших инструментов для успешного взаимодействия. Но в среде проектировщиков по-прежнему нет единого мнения на этот счет, не всегда при создании сайтов и приложений уделяется должное внимание анимации интерфейсов. В связи с этим проблема использования анимации интерфейсов является предметом оживленных дискуссий и научных исследований.

Так, С. Хадсон и Д. Стаско [1] из Технологического института Джорджии, а также Н. Баскандери [2] и другие считают, что анимация снижает когнитивную нагрузку и позволяет увеличить объем воспринимаемой пользователем информации. Многие исследователи отмечают привлечение внимания и соответствие реальному миру как одни из наиболее важных функций анимации и показывают, что анимация смягчает и объясняет переход между разными состояниями интерфейса [3].

Проведенные научные исследования в рассматриваемой области преимущественно касаются использования анимации в дизайне графического интерфейса операционных систем. В то время как проблема анимирования интерфейсов веб-сайтов и мобильных приложений исследована недостаточно, что определяет актуальность темы научной работы.

Интерес современных исследователей к использованию анимации объясняется тем, что главной тенденцией в современном дизайне является отход от скевоморфизма к так называемому «плоскому» дизайну, характеризующемуся отказом от реалистичности в графическом отображении элементов интерфейса. Однако потребность пользователей в соотношении графических объектов с объектами реального мира по-прежнему сохраняется.

Анимация способна улучшить взаимодействие без использования тяжеловесных графических решений. Она создает ощущение работы с объектом реального мира, оставляя минимум интерфейса. К тому же в ориентированном на человека проектом подходе пользовательский интерфейс должен быть интуитивным и отзывчивым. Функциональная анимация помогает достичь этих целей.

Функциональная анимация – это анимация, встроенная в дизайн пользовательского интерфейса, как часть функциональности этого дизайна.

Хорошо продуманная анимация имеет потенциал для выполнения нескольких функций.

– Снижение когнитивной нагрузки. Когнитивная нагрузка – это объем умственных усилий, необходимых для выполнения задачи. Когда пользователи находятся на сайте, они получают большой объем информации для обработки. В результате работает избирательное внимание, а именно, «туннельное видение», т.е. пользователи фокусируются непосредственно на области экрана, над которой они работают, поэтому они не воспринимают всю информацию на экране сразу [2].

Удобный интерфейс должен сокращать умственные усилия пользователя, необходимые ему для выполнения задачи. Функциональная анимация упрощает выбор пользователя, привлекая внимание к наиболее важной информации и объектам.

– Привлечение внимания пользователя. Представители ведущей в области исследований юзабилити компании NNgroup утверждают, что «эффективный способ использования анимации – привлечение внимания пользователя». Анимация позволяет сосредоточить внимание на иерархии и отношениях между экранами и элементами на экране.

Человек замечает движущиеся объекты быстрее, чем неподвижные, что обеспечивает его безопасность. Анимация способна привлекать внимание даже на периферии зрительного поля, и поэтому может быть использована как сильный стимул для фокусировки внимания даже в сложных интерфейсах [4].

Визуальная иерархия. Анимация порядка загрузки элементов страницы на экране может отображать визуальную иерархию. Использование ненавязчивой анимации показывает пользователю структуру страницы и то, на чем следует сосредоточиться, что уменьшает когнитивную нагрузку. Кроме того, благодаря использованию одной и той же анимации для разных элементов пользователю становится понятно, что элементы сгруппированы и могут выполнять подобные действия.

– Обеспечение лучших решений. Продуманная анимация позволяет пользователю быстро понять взаимосвязи информации, в результате чего он может принимать более правильные решения. Так переходы между состояниями могут связывать функциональность одного объекта.

Анимация помогает пользователям создавать ментальные карты пространственной информации. Анимация позволяет пользователям узнать, как использовать приложение или сайт, проследив переходы между экранами. Например, просматривая страницу сайта, пользователь может обнаружить, что для возврата на главную страницу он может прокручивать влево. Для обеспечения полной открытости графическое изображение должно использоваться в сочетании с анимацией.

– Визуальный отклик действиям пользователя. Хороший дизайн интерфейса обеспечивает обратную связь. Обратная связь заставляет пользователя чувствовать, что он взаимодействует с реальными элементами на экране, и демонстрирует результат этого взаимодействия – успешное или неуспешное завершение действия. Элементы пользовательского интерфейса, такие как кнопки и другие элементы управления, должны казаться осязаемыми. Визуализация и движение могут обеспечить осязаемость, изображая ответные действия системы так, чтобы действия пользователя выглядели как непосредственные манипуляции над системой [4].

Визуальная обратная связь также полезна, когда необходимо информировать пользователей о результатах операции. В случаях, при которых операции не выполняются успешно, функциональная анимация предоставляет информацию о проблеме быстрым и легким способом. Например, анимацию встряхивания часто используют при вводе неверного пароля. Это объясняется тем, что тряска – универсальный жест для сообщения слова «нет», поскольку простое качание головы – это то, как люди дают обратную связь друг другу.

– Видимость состояния системы. Эта функция выступает в качестве первой из 10 фундаментальных эвристик юзабилити. Процесс загрузки и выгрузки данных является примером использования функциональной анимации. Анимированные загрузочные полосы показывают, как быстро проходит процесс и как скоро действие будет обработано. Этот тип функциональной анимации также можно использовать для привлечения внимания пользователя к важным изменениям состояния в приложении (или даже в системе), например, при входящем вызове.

– Навигационные переходы. Навигационные переходы – это переходы между состояниями, например, от представления информации высокого уровня до подробного представления или смена элементов на одном уровне иерархии. Навигационные переходы могут быть иерархическими (от родительских к дочерним) или переходами между сестрами. Функциональная анимация помогает увидеть, откуда появился новый объект и где скрывается предыдущий объект, чтобы его можно было найти снова, что создает визуальные связи между состояниями.

Перечисленные функции выполняет только анимация, обладающая такими качествами, как ненавязчивость, естественность, согласованность, чистота. Частота, продолжительность, скорость и доступность анимации также влияют на восприятие пользователем системы, поэтому дизайнеры должны учитывать это при создании анимации [2].

Для достижения необходимых качеств и функциональности анимации используются следующие принципы и приемы [5]:

– Динамика (easing) или движение с ускорением. Динамичное движение выглядит более естественно, чем линейное. Когда динамика движения проработана, пользователи воспринимают движение как естественное и в значительной степени невидимое – что правильно, потому что не отвлекает. Линейное движение слишком заметно и выглядит незавершенным и отвлекающим.

– Смещение и задержка (offset & delay). Определяет отношения объектов и иерархию при введении новых элементов и сцен.

Например, анимацией для трех объектов, где последний появляется с задержкой, можно показать, что два верхних объекта объединены, а нижний объект является отдельным. Используя не только пространство, но и время, можно показать, что иконки или несколько кнопок выполняют разные функции, если они появляются не одновременно, а по очереди.

- Группирование (parenting). Создает пространственные и временные иерархические отношения при взаимодействии с несколькими объектами. Это мощный принцип, который «связывает» объекты в пользовательском интерфейсе. Группирование – это привязка одних свойств объекта к другим свойствам объекта.
- Трансформация (transformation). Создает непрерывный поток повествования при изменении свойств объекта. Трансформация приводит к тому, что происходит когнитивное разделение ключевых моментов в пользовательском опыте на непрерывные серии событий. Эта непрерывность приводит к повышению осведомленности пользователей, запоминанию и последующему использованию.
- Смена значений (value change). Создает динамическое и непрерывное повествование при изменении значений чего-либо. Когда мы используем представления динамических систем в виде значений, основанных на анимации, пользователи понимают, что могут быть причиной изменения этих значений.
- Наложение маски (masking). Создает непрерывность в объекте интерфейса или группе объектов, когда свойство определяет, какая часть объекта или группы отображается или скрывается.
- Перекрытие (overlay). Создает пространственные отношения объектов в визуальной плоскости, когда многослойные объекты зависят от местоположения. Перекрытие позволяет использовать движение, чтобы открывать объекты, которые находятся за или перед другими в двумерном пространстве.
- Клонирование (cloning). Создает непрерывность, взаимосвязь и повествование, когда появляются и уходят новые объекты.
- Параллакс (parallax). Создает пространственную иерархию в визуальной плоскости при прокрутке. Параллакс, как принцип UX в анимации, описывает различные объекты интерфейса, движущиеся с разной скоростью.
- Размерность (Dimensionality). Обеспечивает пространственную структуру повествования, когда появляются и исчезают объекты. Критически важными для пользовательского опыта являются непрерывность, а также пространственная ориентация. Принцип размерности – это способ преодолеть нелогичный пользовательский опыт в двумерном пространстве.
- Приемы Dolly & Zoom. Сохраняет непрерывность и пространственное повествование при навигации по объектам интерфейса и пространству.

Таким образом, основными результатами проведенного исследования являлись обобщенный набор функций анимации интерфейса и качеств функциональной анимации, а также принципы применения анимации для повышения качества UX, предложенные отечественными и зарубежными исследователями.

Литература

1. Hudson S., Stasko J.T. Animation Support in a User Interface Toolkit: Flexible, Robust, and Reusable Abstractions // Proceedings of the 6th annual ACM Symposium on User Interface Software and Technology. – 1993. – P. 57.
2. Baskanderi N. UI Animation: Use Responsibly [Электронный ресурс]. – Режим доступа: <https://uxdesign.cc/ui-animation-please-use-responsibly-e707dbdb12d5> (дата обращения: 02.12.2017).
3. Беляев А.А. Анимация в дизайне интерфейса информационных сайтов [Электронный ресурс]. – Режим доступа: <http://www.mediascope.ru/1618#5>, своб.
4. Babich N. How Functional Animation Helps Improve User Experience [Электронный ресурс]. – Режим доступа: <https://www.smashingmagazine.com/2017/01/how-functional-animation-helps-improve-user-experience/>, своб.
5. Willenskomer I. Creating Usability with Motion: The UX in Motion Manifesto [Электронный ресурс]. – Режим доступа: <https://medium.com/ux-in-motion/creating-usability-with-motion-the-ux-in-motion-manifesto-a87a4584ddc>, своб.



Головатая Елена Николаевна

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4205

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: enchumakova@mail.ru

УДК 004.92

ИССЛЕДОВАНИЕ ОСОБЕННОСТЕЙ UI/UX ДИЗАЙНА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Головатая Е.Н.

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

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

Работа посвящена исследованию особенностей UI/UX дизайна мобильных приложений. Рассмотрены понятия UI и UX дизайна. Произведен анализ интерфейса мобильных и экранных устройств. Сделан обзор на ограничения мобильного формата. Рассмотрены общие правила дизайна взаимодействия Нильсена. В работе описаны основные принципы проектирования мобильных интерфейсов.

Ключевые слова: UI и UX дизайн, мобильное приложение, проектирование интерфейсов, тренды UI дизайна, принципы проектирования, пользовательский опыт.

UX (User eXperience) с английского переводится как опыт взаимодействия, он включает в себя совокупность всех впечатлений и ощущений пользователя до, во время и после взаимодействия с интерфейсом. В этом термине заложены психологические и поведенческие аспекты взаимодействия пользователя с программными продуктами.

UI – это User Interface (дословно «пользовательский интерфейс») – то, как выглядит интерфейс и то, какие физические характеристики приобретает. UI-дизайн определяет цвета, типографику и другие видимые физические характеристики [1].

В целом эти два понятия являются проектированием любых пользовательских интерфейсов, в которых удобство использования также важно, как и внешний вид. Разработанный по всем правилам пользовательский интерфейс значительно повышает эффективность продукта и дает ему конкурентные преимущества.

Существует множество устройств с микроотличиями в размерах и эстетике. Нет смысла проектировать уникальный интерфейс для каждой модели. Зато все устройства можно условно разделить по типу места использования – стол, диван и карман, поэтому при создании интерфейса необходимо учитывать отличия мобильных и настольных версий экранов.

Например, организация экрана на настольном устройстве сделана произвольно, элементы пользовательского интерфейса должны быть расположены в поле зрения. А элементы пользовательского интерфейса на мобильных версиях должны быть расположены в соответствии ограничения пользования одной рукой [2].

В целом мобильные системы должны иметь различные руководящие принципы и принципы процесса проектирования. Однако можно выделить основной – принцип одного контекста. Минимизация количества контекстов привело к тенденции взаимодействовать с контентом без посредника, без нависающих окон, без смены контекста, поэтому, например, все картинки начали нажиматься и увеличиваться.

Мобильный формат имеет как сильные стороны, так и ограничения. Одно из главных ограничений – маленькие размеры экрана. Приоритет содержания и функций является ключевым фактором, это означает, что необходимо предоставлять элементы пользовательского интерфейса для работы над текущим контентом. Еще один важный признак мобильных устройств – их портативность, а это значит – прерываемость. При создании интерфейсов необходимо предусматривать сохранение контекста и сделать его легким и сходу понятным для пользователей, чтобы быстро восстановить контекст и возобновить прерванную задачу. Также нужно учитывать, что дизайн должен быть самодостаточным: любые мобильные составные задачи должны быть легко завершаемы в одном приложении или на одном сайте. Пользователи не должны оставить приложение, чтобы найти необходимую информацию о том, что требует и не предоставляет приложение. Жесты представляют собой скрытый, альтернативный пользовательский интерфейс, который делает взаимодействия подвижными и эффективными и может сохранить полезную площадь экрана.

При проектировании удобных пользовательских интерфейсов зачастую пользуются общими правилами дизайна взаимодействия, разработанными в 1995 году Якобом Нильсеном, которые используются и по настоящий день. Они называются «эвристиками», потому что являются широкими эмпирическими правилами, а не руководящими принципами удобства использования. К тому же, большинство существующих принципов являются частными случаями общего, нет смысла рассматривать каждую «новую» модель, поэтому имеет место обобщение [3]:

1. видимость состояния системы или правило обратной связи;
2. метафора или соответствие между системой и реальным миром;
3. свобода действий пользователя и управляемость;
4. единообразие и стандарт;
5. предотвращение ошибок;
6. понимание лучше, чем запоминание;
7. гибкость и эффективность использования;
8. эстетичный и минималистичный дизайн;
9. распознавание и диагностирование ошибок;
10. справка и документация;
11. один контекст;
12. человечность.

UI-дизайн для мобильного приложения отличается от дизайна пользовательского интерфейса для веб-сайтов. В работе над дизайном мобильных приложений стоит использовать некоторые стандартные решения, потому что ряд принципов, позволяющих создавать привлекательные интерфейсы, которые нравятся пользователям, уже наработан.

Устройство, на котором установлено приложение, обладает функциями, которые могут усиливать функциональность мобильного приложения. При создании дизайна интерфейса это нужно учитывать, используя, к примеру, камеру, GPS и другие функции. Также следует принимать во внимание операционную систему, установленную на мобильном устройстве – если это Android, то нужно пользоваться гайдлайном MaterialDesign, а если это iOS, следовать рекомендациям дизайнеров Apple [4].

Сейчас нужно создавать интерактивные и удобные интерфейсы, при этом принимать во внимание следующие принципы:

- в интерфейсе должна быть понятная кнопка, позволяющая переходить к следующему действию;
- интерактив должен поддерживаться девайсом и мобильной платформой;
- нужно использовать унифицированные иконки и различные символы;
- в каждом экране приложения должна присутствовать только актуальная информация;
- обратную связь от пользователей приложение должно получать в режиме реального времени.

Создавая дизайн мобильного приложения нужно делать акцент на содержании. Контент всегда должен быть на первом месте. Он должен быть представлен наиболее оптимальным образом. К примеру, если приложение продает интерактивные экскурсии, то процесс прохождения экскурсий должен быть оптимизирован таким образом, чтобы пользователю были понятны и доступны все его предыдущие и последующие шаги. Также не стоит забывать об интерактивных элементах дизайна. Данные элементы должны точно идентифицироваться пользователем. Все основные элементы навигации должны присутствовать на каждом экране приложения. Например, если приложение представляет собой ленту, можно при прокручивании вниз через некоторое время прятать основной блок навигации, а при прокручивании вверх возвращать. Таким образом, пользователь в любой момент может вернуться к навигации.

В данной работе были определены главные ограничения мобильных платформ. К ним относятся: маленький размер экрана, прерываемость, единое окно, сенсорный экран. Было установлено, что в результате мобильности нужно учесть принцип одного контекста при проектировании мобильного интерфейса. На основе процесса UX принципы проектирования мобильных интерфейсов данного проекта также находятся под влиянием потребностей конечных пользователей. Для того чтобы удовлетворить потребности пользователей, интерфейсу придется стать «человечным». Вот что это значит:

- запоминать привычки человека;
- учитывать контекст и сообщать об отклонениях;
- решать проблемы самостоятельно;
- поддерживать черновики и промежуточные состояния процесса;
- быть незаметным;
- подстраивать элементы управления контентом под каждый процесс.

Литература

1. Что такое UX и UI дизайн – Особенности и отличия [Электронный ресурс]. – Режим доступа: <https://www.kasper.by/blog/что-такое-ux-i-ui-dizain/> своб.
2. Гарретт Д. Веб-дизайн: книга Джесса Гарретта. Элементы опыта взаимодействия. – СПб.: Символ-Плюс, 2008. – 64 с.
3. Шепелёв М.Д., Флеров А.В. Совершенствование юзабилити-тестирования интерфейса // Наука, образование, общество: актуальные вопросы и перспективы развития: сб. научных трудов по материалам Международной научно-практической конференции: в 3 частях. – Ч. 2. – 2015. – С. 63–64.
4. Krishna G. The Best Interface Is No Interface: The simple path to brilliant technology (Voices That Matter). – Изд-во: Питер, 2016. – Р. 123–136.



Горбунов Антон Вячеславович

Год рождения: 1992

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4205

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: korvin_mckernek@mail.ru

УДК 004.92

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СЕРВИСОВ ОБЛАЧНОГО РЕНДЕРИНГА

Горбунов А.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 определено понятие «облачные вычисления», приведена классификация облачных вычислений по модели обслуживания, представлены результаты сравнительного анализа облачных сервисов рендеринга.

Ключевые слова: компьютерная графика, рендеринг, 3D-визуализация, облачные сервисы для рендеринга, облачные вычисления, модели облачных услуг.

Необходимость обращения к сервисам удаленной 3D-визуализации обусловлена проблемой нехватки вычислительной мощности персональных компьютеров при рендеринге тяжелых детализированных сцен. Покупать дополнительное оборудование порой не могут позволить себе не только отдельные специалисты, но также и компании, вовлеченные в проекты в области трехмерного проектирования.

На время выполнения рендеринга влияет множество факторов [1]:

- используемые алгоритмы (глобального освещения, сглаживания, затенения и др.);
- уровень детализации визуализируемой сцены (количество объектов в сцене и общее число полигонов, из которых состоит каждый объект);
- типы источников освещения, их характеристики, количество, расположение;
- используемые материалы и их свойства, текстуры, визуальные эффекты;
- размеры, разрешение, формат и количество частей выходного изображения.

Таким образом, с ростом требований к качеству визуализации и с увеличением сложности проектов, при учете всех возможных факторов, в несколько раз повышаются и требования к мощности оборудования, а вместе с тем и возрастает время рендеринга, поэтому финальный рендеринг является очень ресурсоемкой задачей. Облачные вычисления в этом случае могут решить вопрос нехватки ресурсов.

Согласно стандарту NIST SP 800-145, The NIST Definition of Cloud Computing, разработанному в Институте Стандартов и Технологий (NIST) США, облачными вычислениями называется «модель предоставления повсеместного, удобного сетевого доступа по требованию к общему пулу настраиваемых вычислительных ресурсов (например, сетей, серверов, хранилищ данных, приложений, сервисов), которые могут быть получены в собственное распоряжение при минимальных управленческих усилиях или взаимодействиях с поставщиком услуги» [2].

По модели предоставления услуг облачные вычисления и сервисы, построенные на базе облачных вычислений, принято разделять на следующие категории [2]:

1. IaaS (Infrastructure-as-a-Service) – инфраструктура как услуга. На данном уровне потребитель обладает возможностями построения собственной облачной инфраструктуры и ее управлением;
2. PaaS (Platform-as-a-Service) – платформа как услуга. Потребитель облачных услуг получает возможность и средства для самостоятельного создания, тестирования и эксплуатации

программного обеспечения (ПО), а задачи управления информационной инфраструктурой, включая вычислительные сети, серверы и системы хранения, решаются провайдером;

3. SaaS (Software-as-a-Service) – ПО как услуга. На этом уровне провайдер предоставляет пользователям «облака» готовое веб-приложение, доступ к которому, как и к хранящимся в «облаке» данным, осуществляется, как правило, через мобильное приложение или веб-браузер.

Исследование рынка сервисов облачного рендеринга для взаимодействия с собственным ПО для трехмерного моделирования является одной из актуальных на сегодняшний день задач для компании АСКОН.

Компания является российским разработчиком и интегратором инженерного ПО. Одним из направлений работы компании является разработка системы автоматизированного проектирования (САПР) для архитектурного и промышленного проектирования, среди которых к трехмерному моделированию относятся [3]:

1. КОМПАС-3D – система трехмерного моделирования для проектирования трехмерных моделей в различных сферах производства;
2. Renga Architecture – BIM-система для архитектурно-строительного проектирования;
3. Renga Structure – BIM-система для проектирования конструктивной части зданий/сооружений.

Выполнение визуализации трехмерных моделей требуется для проектов, выполненных в первых двух САПР.

Выбор онлайн-сервисов для рендеринга в «облаке» 3D-моделей, созданных в ПО АСКОН, осуществлялся на основании следующих критериев:

1. модель обслуживания: SaaS. Такой выбор обусловлен тем, что при использовании SaaS-решений пользователю не нужно производить установку и настройку необходимого ПО. Все что требуется от сервиса – возможность загрузить необходимые файлы, произвести настройки рендеринга, запустить процесс и получить результат в удобном формате;
2. поддержка выходных форматов 3D-файлов КОМПАС-3D и Renga Architecture, указанных в табл. 1.

Таблица 1. Выходные форматы 3D-файлов ПО АСКОН для трехмерного моделирования

САПР	Форматы выходных файлов
КОМПАС-3D	STEP, IGES, STL, DXF, DWG, PARASOLID, VRML
Renga Architecture	OBJ, COLLADA, STL, 3DS, LWO

Сформированным критериям (с некоторыми замечаниями) удовлетворили сервисы, данные о которых приведены в табл. 2.

Таблица 2. Сводная таблица с данными о сервисах

Сервис	Условия использования		Количество поддерживаемых форматов		Фотореализм
	Минимальный тариф	Ограничения тарифа	КОМПАС-3D	Renga Architecture	
Autodesk Rendering	Подписка на ПО	100 кредитов	7/7	3/5	Да
OneRender	Free	50 кредитов, 1 GPU, 2GB, HD-качество изображения	1/7	3/5	Недостаточно фотореалистично
Modelo.io	Free	Макс. вес проекта: 500MB, макс. вес модели: 200MB	1/7	3/5	Нет
Autodesk Viewer	Free Trial	100 кредитов на год, 5GB	6/7	4/5	Нет
Clara.io	Basic/Free	5GB, ∞ к-во открытых	7/7	5/5	Да

Сервис	Условия использования		Количество поддерживаемых форматов		Фотореализм
	Минимальный тариф	Ограничения тарифа	КОМПАС-3D	Renga Architecture	
		сцен, поддержка на форуме			

Перечисленные в табл. 2 сервисы были подвергнуты сравнительному анализу по ряду признаков: количество поддерживаемых форматов 3D-файлов ПО АСКОН, наличие возможности выполнять фотореалистичную визуализацию моделей, ограничения при минимальных тарифах.

Из таблицы видно, что все сервисы, кроме Autodesk Rendering, позволяют пользователю бесплатно опробовать возможности рендеринга. Но практически в каждом из них рендеринг производится в обмен на специальные облачные единицы (или кредиты), определенное количество которых изначально входит в пакет услуг минимальных тарифов рассмотренных сервисов.

Следует отметить следующие недостатки сервисов:

1. возможность отправлять 3D-модели в «облако» Autodesk Rendering можно только из установленных на локальном компьютере продуктов Autodesk: AutodCAD, Revit, Fusion 360, 3ds Max и Navisworks. Количество и виды поддерживаемых форматов также зависят от установленных программ;
2. визуальная оценка примеров рендеринга с помощью сервиса OneRender не позволяет назвать результаты фотореалистичными. Изображения, полученные по завершении рендеринга, являются фотореалистичными тогда, когда глаз не может определить, что находится перед ним: фотография или визуализированная 3D-сцена [4]. Результаты рендеринга в OneRender подобным качеством не обладают;
3. Modelo.io и Autodesk Viewer представляют собой онлайн-инструменты для интерактивного просмотра трехмерных моделей в облаке и не обладают возможностью выполнять рендеринг в облаке, однако, в первом случае предоставление услуг рендеринга возможно по предварительной договоренности;
4. базовый тариф Clara.io предполагает, что все проекты будут доступны для просмотра в открытом доступе, что не является допустимым в отношении коммерческих разработок.

Результаты сравнительного анализа и перечисленные недостатки сервисов позволили сделать вывод, что наилучшим из рассмотренных сервисов, который можно порекомендовать для осуществления задач рендеринга в «облаке», является Clara.io. Помимо того, что сервис поддерживает все форматы ПО АСКОН для трехмерного моделирования и позволяет выполнять высокореалистичную визуализацию с помощью системы визуализации V-Ray, данное SaaS-приложение предоставляет также возможности для реализации всех этапов создания компьютерной 3D-графики, а не только рендеринг, благодаря чему Clara.io можно по праву назвать online-САПР.

Литература

1. Лысыч М.Н. Визуализация 3D-объектов в системах автоматизированного проектирования // Международный журнал прикладных и фундаментальных исследований. – 2016. – № 11(5). – С. 853–857.
2. NIST SP 800-145, The NIST Definition of Cloud Computing [Электронный ресурс]. – Режим доступа: <http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>, своб.
3. Сайт компании АСКОН [Электронный ресурс]. – Режим доступа: <https://ascon.ru/>, своб.
4. Горбунов А.В. Анализ методов рендеринга моделей в системе рендеринга V-Ray для достижения фотореалистичности моделей // Альманах научных работ молодых ученых Университета ИТМО. – 2017. – Т. 2. – С. 7–10.



Давлетшин Владислав Равилевич

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4208

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: vlad35066@mail.ru

УДК 004.9

К ВОПРОСУ О ПРОБЛЕМАХ И ОШИБКАХ, СУЩЕСТВУЮЩИХ В ТЕКУЩИХ JAVASCRIPT-ФРЕЙМВОРКАХ

Давлетшин В.Р.

Научный руководитель – д.т.н., профессор Погорелов В.И.

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

В работе рассмотрены проблемы, связанные с работой текущих популярных front-end JavaScript-фреймворках. Рассмотрены примеры возможных решений некоторых проблем. Проведен анализ этих проблем, откуда они появились, приведены примеры – как этих проблем можно было избежать.

Ключевые слова: фреймворк, разработка, front-end, JavaScript, ошибки, документация.

Популяризация деятельности web-dev сообщества позволила привлечь огромное количество людей в эту сферу и впоследствии внедрить огромное количество новых технологий. Появление на рынке технологий, таких как jQuery, ECMAScript5-6, AJAX и прочих, позволяющих использовать DOM (Document Object Model) [1] с большей эффективностью – стало огромным толчком для производства разного рода фреймворков и библиотек. Другие, в свою очередь, открывают перед разработчиком огромный потенциал для работы и составления приложения.

Вместе с ростом сообщества, ростом количества технологий и ростом спроса в веб-сфере, появляются также огромные проблемы в функционале или работе имеющихся технологий, некоторые – закрывают дыры в функциональности новыми версиями, есть такие, которые прекращают поддержку возможностей, с которыми их продукт не смог справиться корректно. Но неизменно остается одно, ошибки и проблемы в технологиях будут всегда, и в этой работе изложены самые распространенные ошибки и проблемы, с которыми может столкнуться рядовой пользователь при работе с этими технологиями.

1. Angular. Ангуляр – это, очевидно, плохой выбор. Он устарел, и новые проекты ни в коем случае не должны использовать его. Ангуляр появился из ниоткуда и стал лучшим по умолчанию, благодаря прорывному дизайну, является одним из первых сильных фреймворков JavaScript. Многие проблемы, которые создал сам для себя Ангуляр – это терминология, которой он ввел слишком много и множество странных архитектурных решений. Например, \$scope, непонятные директивы, не до конца продуманная трансклюзия. Ангуляр также делает некоторые вещи, которые в корне неверны, например, создание недопустимой разметки HTML. Даже Google не использует Ангуляр для своих собственных приложений, таких как Gmail, и для этого есть причина.

Его идиосинкразический код, отличающийся от других фреймворков, выглядит незнакомым для всех, кто не обладает большим опытом разработки с его сложностями.

2. Angular2. Ангуляр2 практически не отличается от Ангуляра. Главный недостаток этого фреймворка, что на него переходят многие разработчики после первой версии Ангуляра, не рассмотрев другие (возможно лучшие) варианты. Большинство решений этого

фреймворка – просто необъяснимы, использование Case Sensitive [2] HTML, который вопиюще нарушает стандарты HTML и требует реализации собственного парсера, шаблоны – «синтаксическая каша», которая не имеет никакого смысла никому вне собственной системы Angular2.

```
<li * ngFor = "let article for articles">  
<a [routerLink]="['/article', article.id]"> {{ article.title }} </a>  
</ li>
```

Любой разработчик JavaScript взглянув на приведенный фрагмент кода будет озадачен. Что за аргумент со звездочкой? Квадратные скобки – это массив? Одно ли значение по разные стороны присваивания?

TypeScript – еще один необъяснимый выбор, ведь малое количество разработчиков его знают по сравнению с JSNative, почему было принято решение не писать на языке уже известном, а заставлять сообщество изучать новый?

3. Aurelia. Самый большой недостаток Аурелы на рынке – непопулярность. Ее мало кто использует, соответственно, из-за маленького сообщества – небольшое количество готовых решений, искусственное повышение порога вхождения из-за маленького количества решенных вопросов на StackOverflow.

Но это совсем не значит, что это единственная проблема. Выбор JSPM [3] как сборщик – наверное, является одним из худших решений, он не поддерживается никаким другим проектом в отличие от более стандартных сборщиков: Webpack, Gulp и подобных.

И, конечно, не самая большая, но определенно самая важная проблема – сказать, что у Аурелы маленькая документация – это приуменьшение. Решение некоторых ошибок можно найти на сторонних ресурсах, таких как StackOverflow, но в документации их так и не увидеть.

4. React. Проблема в том, что сам Реакт фактически не решает какой-либо реальной проблемы. Это не фреймворк, а библиотека. Если все, что вам нужно, это компонентная ViewModel, то все идеально, но это не так. Чтобы решить любые проблемы, которые не являются тривиальными, вам придется создать неудобный стек на Реакте, своего рода маршрутизатор, шаблон состояний Redux, Flux-для собственных проверок и событий, это все нужно еще подвязать с помощью Babel Webpack, чтобы сделать это в ES6.

В итоге, чтобы успешно разрабатывать на Реакте, из-за того, что это не фреймворк, а библиотека – вам придется самому создать фреймворк, однако стабильность оставляет желать лучшего, все компоненты не имеют единой основы и балансируют в стеке друг на друге. Из-за этой проблемы вам также нужно поддерживать сборку и все более сложный API.

Это конечно не значит, что Реакт в отдельности безупречен. К примеру, постоянная необходимость добавления предложения WHERE, при ручном обновлении SQL-таблицы перед ее развертыванием. Множество мелких улучшений, которые, как говорит сообщество, «просто не применимы к Реакту», на самом деле были бы вполне успешны. Разделение проблем и контроль качества кода, тестирование кода, завершение кода и подсветка синтаксиса также важна.

И не закрытым вопросом остается лицензия на использование продукта, изначальная формулировка была такова «Facebook может изменить формат вашей лицензии» или «пересмотреть условие использования лицензии», однако теперь вам в принципе могут запретить использование Реакта, конечно, такое случается очень редко и вряд ли случится, но такая возможность остается.

Реакт был прорывом, самым быстрым фреймворком, стал первым использовать Virtual DOM, заставив другие платформы конкурировать, и у них получилось лучше, например, даже Vue2 и Inferno бьют рекорды Реакта по производительности.

5. Ember. За последние несколько лет Эмбер потерял позиции по популярности. С небольшой популярностью – небольшой спрос, лишь некоторые компании используют его.

Одна из основных проблем Эмбера – у него на все есть свое решение, это явно видно даже в деталях, у него собственный сборщик «Брокколи» вместо стандартов в наше время, таких как Webpack. Несмотря на строгий шаблон MVC – он не требует контроллеров, а иногда требует, вместо роли контроллера здесь собственное нововведение-маршрут. Но здесь есть и контролер, туда можно поместить и события, и действия, но возможно они не будут корректно выполняться, и придется перенести их обратно в маршрут.

Эмбер – самый что ни на есть «фреймворкнутый» фреймворк, и если в них есть проблемы, то в Эмбере их больше всего. Вес фреймворка – Эмбер самый большой, навязывание мнения – вспомните «Брокколи» и маршруты, когнитивная нагрузка – огромный API, производительность – Эмбер самый медленный.

В теории есть, разумеется, и хорошие идеи. Например, согласование настроек, но у Эмбера нет достаточной гибкости, его согласования слишком строго соблюдаются и могут быть изменены только с помощью его CLI. CLI – тоже хорошо, такой инструмент становится все более популярным и используется в других фреймворках, только он является удобной функцией, а опять же из за гибкости в Эмбере он обязателен.

Эмберу не хватает поддержки, которую Ангуляр получает от Google, а Реакт от Facebook, однако Эмбер все еще используется для «амбициозных» приложений. Им пользуются огромные компании, но, если у вас нет такого масштаба и таких ресурсов, вам, вероятно, не нужна такая сложная всепоглощающая структура.

Заключение. В заключение стоит сказать, что анализ технологий был не глубоким, а в работе выделены лишь проблемы и ошибки, которые являются частью ключевых проблем в современных фреймворках. В итоге становится ясно, что большинство самых больших проблем, связанных с фреймворками – это уход от стандартов программирования в текущий момент времени – как появление «Брокколи» в Эмбере, так и появление TypeScript в Ангуляре 2. Чем дальше продвигаются технологии, тем больше проблем с ними возникает. Множество более важных ошибок, которые существовали в перечисленных технологиях и не были упомянуты, были исправлены в новых версиях, что свидетельствует тому, что технологии даже с маленьким охватом аудиторией постоянно исправляются и развиваются.

Литература

1. DOM [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Document_Object_Model, своб.
2. Is AngularJS is case sensitive? [Электронный ресурс]. – Режим доступа: <https://www.quora.com/Is-AngularJS-is-case-sensitive>, своб.
3. Jspm vs Webpack [Электронный ресурс]. – Режим доступа: <https://ilikekillnerds.com/2015/07/jspm-vs-webpack/>, своб.
4. React: The Virtual DOM [Электронный ресурс]. – Режим доступа: <https://www.codecademy.com/articles/react-virtual-dom>, своб.



Деркунская Светлана Александровна

Год рождения: 1996

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: svetlana7-127@mail.com



Шуклин Дмитрий Анатольевич

Год рождения: 1979

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: do@limtu.ru

УДК 003

**ФОРМООБРАЗУЮЩИЕ ФАКТОРЫ ДИНАМИЧЕСКОЙ АЙДЕНТИКИ –
ТИПОГРАФИКА И ВИЗУАЛЬНЫЙ ОБРАЗ**

Деркунская С.А., Шуклин Д.А.

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

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

В работе рассмотрены: использование визуальной айдентики в современном дизайне логотипов и фирменного стиля, основные формообразующие факторы динамической айдентики – типографика и визуальный образ, а также методики создания динамической айдентики.

Ключевые слова: динамическая айдентика, типографика, визуальный образ, логотип, фирменный стиль, dynamic identity, typography, visual image, logo, corporate identity.

За последние несколько лет произошли радикальные изменения в сфере визуальных коммуникаций. Графический дизайн в большей степени стал трехмерным, цифровым и эмоциональным, все меньше модульным, конструктивным и рациональным. В попытке улучшить визуальный образ, при этом учитывая когнитивные тенденции пользователей и аспекты социальных сетей, у дизайнеров появилась идея внести изменения в графику, сделав ее многомерной, вариативной и постоянно трансформирующейся.

Так, в графическом дизайне появилось такое направление как динамическая (Dynamic) или генерирующая айдентика (Generative Identity).

На сегодняшний день в среде дизайнеров принято отождествлять термины «визуальная айдентика» и «фирменный стиль». Когда как на деле «айдентика» (identity) – более обширное понятие и предполагает за собой не только визуальные идентификаторы, но и словесные константы (слоганы), принципы психологического восприятия бренда потребителем, «психологический образ» компании, набор принципов визуальной презентации бренда на рынке, «лицо» компании в глазах клиента.

В деятельности зарубежных стран термин «айдентика» носит название «corporate identity» – «корпоративная идентичность» или «brand identity» – брендинг [1]. В России визуальный образ компании всегда называли корпоративной идентичностью, и только в последнее время это понятие заменил модный термин «айдентика».

Визуальная айдентика (от англ. visual identity – визуальная идентичность; зрительный образ) – это визуальная составляющая бренда, призванная повысить его узнаваемость и создать впечатление целостности [2].

Айдентика играет важную роль в развитии компании. Она является связующим звеном между основной аудиторией потребителей и торговой маркой. Айдентика выполняет важные функции:

- помогает идентифицировать компанию, ее продукты из основной массы конкурентов;
- привлекает внимание, формирует отношение деловых партнеров и потребителей на эстетическом уровне;
- способствует снижению затрат на продвижение новых продуктов бренда;
- повышает эффективность маркетинговых мероприятий, рекламных компаний;
- помогает сформировать положительный имидж организации [3].

Идентификация бренда приобретает все больше личностный, человеческий смысл. Айдентика вынуждена становиться гибкой, органичной той среде, для которой она предназначена.

Оля Дивисенко в своей статье «Динамическая айдентика» [4] выделяет ее главное отличие: «Главное отличие динамических логотипов – это изменения внутри айдентики в самых разных проявлениях. Правил как таковых нет, но есть условие: как бы ни менялись ее составляющие, динамическая айдентика должна выполнять свою главную задачу, такую же, как у обычной, – быть узнаваемой. Глядя на компании, которые используют динамическую айдентику, нельзя не заметить, что в большинстве своем – это творческие, культурные, инновационные или образовательные проекты. Но на самом деле есть множество удачных примеров в абсолютно разных сферах бизнеса, в том числе далеких от творчества».

Модульная сетка, типографика, фирменная палитра, узор – все это составляющие визуальной системы, которые способны меняться, более того их изменчивость и становится залогом устойчивости всей графической системы. Играя цветом или изображением, можно создать невероятное разнообразие, оставаясь при этом узнаваемым.

Одним из первых, кто внедрил динамическую айдентику в иностранные проекты, стал Manhattan Design, создавший логотип для MTV в 1981 году. Они создали систему, в которой «М» и «TV» имеют фиксированную форму и положение, однако, могут быть построены бесчисленным многообразием вариаций цветов, узоров, текстур, анимаций и иллюстраций.

В дальнейшем этот метод получил название «Container» (Контейнер) – представление логотипа в качестве контейнера, наполнение которого постоянно меняется. Выбрав одну характеристику – цвет, набор картинок, – открывается бесчисленное количество узнаваемых вариантов логотипа. Сегодня эта система содержимого контейнера по-прежнему популярный способ создания динамичной идентичности бренда.

Wolff Olins создал динамический логотип для AOL, используя похожую систему динамической идентификации, размещая фоновый рисунок за фиксированным элементом. Метод получил название «Wallpaper» (Обои/фон) – изменение фона, на котором расположен логотип. Общая форма может меняться, но образ остается целостным и узнаваемым [1].

Компания AOL успешно перестроилась как лидирующая брендовая компания, вкладывая в контент, который привлекает аудиторию на местном и глобальном масштабах (рис. 1, а).

Следствием попытки применения динамической айдентики все большего количества брендов является стремление соответствовать стандартам именно этим глобальным корпорациям – AOL и Google, выгодно использующих динамическую визуальную айдентику.

Созданный в 2007 компанией Lava логотип IDTV (рис. 1, б) служит примером еще одной структуры динамической идентификации – DNA (ДНК) [1], путем соединения между собой различного набора компонентов. Используя разнообразные вариации, соединения главных элементов каждый раз приводит к новому объекту общей динамической системы.



Рис. 1. Логотип крупнейшей веб-компании Америки AOL (а); логотип компании IDTV (б)

Айдентика – это система. В ней не обязательно должны быть определенные элементы, которые комбинируются между собой. Главное – это язык, формула, по которой она взаимодействует с потребителем. Этот метод получил название «Formula» (Формула) [1] (рис. 2).



Рис. 2. Логотип компании Management for Design

Динамические логотипы уже становятся тенденцией. Их главная цель – эффектно выделить компанию в ее конкурентной среде с помощью ярких визуальных образов и увеличить лояльность покупателей, узнаваемость фирмы. Чем больше компонентов узнаваемы, тем более понятной становится система. Но в рамках этих ограничений, безусловно, есть возможности для еще большего выражения.

Литература

1. Белинский Я. Динамическая айдентика. Часть 1 [Электронный ресурс]. – Режим доступа: <http://www.yaroslavbelinsky.com/2013/01/1.html>, своб.
2. Van Nes I., Hughea P., Van Drunen C. Dynamic Identities: How to Create a Living Brand. – BIS Publishers, 2012. – 48 p.
3. Трофимов Я. Система визуальной идентификации: альтернатива фирменного стиля, правила пользования. – М.: Grey Matter, 2014. – 512 с.
4. Дивисенко О. Динамическая айдентика [Электронный ресурс]. – Режим доступа: <http://blog.galagan.ua/2016/02/02/>, своб.



Дмитриевская Алёна Александровна

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: dmitalyona@yandex.ru

УДК 004.922

ВЕБ-КАРТОГРАФИЯ: СОСТОЯНИЕ И ПЕРСПЕКТИВЫ РАЗВИТИЯ

Дмитриевская А.А.

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 этапы развития веб-картографии и перспективы ее развития. Также приведена классификация основных видов картографических решений и области их применения.

Ключевые слова: веб-картография, карта, картографический интерфейс, API, веб-среда.

Совершенствование, быстрое распространение и повсеместное использование веб-технологий побуждает все большее количество людей использовать современные средства предоставления информации, в том числе и пространственные.

Стоит отметить, что факт возникновения карт в глобальной сети предрешал проектирование и разработку не только особого программного обеспечения (ПО), но и соответствующих аппаратных средств. Такое программное обеспечение стали называть Web Map Service (картографический веб-сервис). С английского языка Web Map Service можно перевести как стандартный протокол для обслуживания через Интернет географически привязанных изображений, генерируемых картографическим сервером на основе данных из базы данных геоинформационной системы (ГИС) [1].

Первые предпосылки проектирования веб-картографических сервисов были инициированы еще в 1990-х годах [2]. Так, например, в 1993 году был разработан и введен в эксплуатацию веб-сервис Xerox PARC Map Viewer. Его основная идея заключалась в отправке запросов прямо от клиента (в основном с использованием браузера) на сервер и получении ответов в виде фрагментов карт, являвшихся изображениями в формате GIF [3].

Минус подобных реализованных в то время веб-сервисов был в узкой тематической направленности и ограничением в локализации передаваемых данных, что не позволяло популяризировать их глобальное использование.

Решающий шаг в решении этого вопроса впервые предприняла Великобритания. В 1998 году был представлен сайт www.streetmap.co.uk, который отличался от предыдущих сервисов: он охватывал всю территорию Великобритании и представлял собой отображение топографических данных. Такое направление предопределило ход моментального распространения этого сервиса среди тысячи людей: для того чтобы найти месторасположение любого объекта на карте, достаточно было знать почтовый индекс, по которому и осуществлялся поиск [4].

Помимо нарастающей популярности использования всевозможных картографических сервисов, возникают и первые инструменты их реализации. Так, в том же 1998 году появляется первое ПО «out of the box», с помощью которого любой пользователь сети имеет возможность создать собственную веб-ГИС – Mapserver [4].

Решающим событием стал запуск глобальных картографических сервисов Google Maps и Google Earth в 2005 году, которые охватывали весь мир. Именно они разрешили проблему узкой локализации данных предшествующих программных средств. Google применила в решении данного вопроса новейший подход в организации сервиса: вместо отправления запроса на сервер, ожидания обработки и получения обратно сгенерированной картинки, подготовка и обработка данных осуществлялась заранее, а сочетание с подходом AJAX позволило мгновенно ускорить работу с картами.

В настоящее время количество картографических веб-решений, позволяющих оперировать пространственными данными, постоянно растет. Появляются новые сервисы, программные решения, интерфейсы, требующие упорядочивания знаний, связанных с вопросами их применения. Как следствие, возникают классификации подобных решений, которые можно рассмотреть исходя из их упорядочивания по обобщающим признакам (таблица).

Таблица. Классификация картографических решений

Наименование	Описание	Примеры
Виртуальные глобусы	Подобные инструменты позволяют быстро реализовать и опубликовать картографические данные в сеть, так как они обрели на рынке массовость и популярность в использовании благодаря быстрому доставлению данных конечному пользователю. Клиентская сторона в данном случае может быть представлена не только браузером, но также и отдельным приложением. При работе с большими объемами данных могут возникнуть трудности с их обработкой, параметризации, анализе	Google Maps, Google Earth, Яндекс.Карты, 2ГИС, Microsoft Visual Earth, ArcExplorer
Пользовательские ГИС	Внутреннее взаимодействие данной группы картографических решений можно рассмотреть с двух сторон: с одной, они являются клиентскими приложениями, позволяющими осуществлять доступ к веб-серверу, с другой – они подготавливают и анализируют данные перед их публикацией в сеть	ArcGIS, Quantum GIS, MapInfo
Картографические веб-сервера	Частные программные продукты, позволяющие спроектировать интерфейс любой сложности, а также произвести интеграцию сервиса с баз данных, которая поддерживает классы пространственных данных (MySQL, PostgreSQL, SQL Server, ArcSDE). Данные картографические решения отличаются от аналогичных наличием полного контроля над ПО и самими данными. К сложностям использования следует отнести непростую установку и настройку, которую можно успешно осуществить, обладая начальными знаниями программирования (например, js или PHP), а также основами администрирования	OpenLayers, GeoServer, MapServer

На основе характеристик обозначенных групп, можно выделить следующие тенденции современных картографических решений:

- предоставление клиенту заранее обработанных данных;
- возможность персонализации и кастомизации сервисов;
- интеграция собственных данных и сторонних сервисов с существующими картографическими решениями;
- глобализация сервисов;

– массовое использование сервисов в повседневной жизни.

В связи с широким распространением картографических сервисов и все более частым применением их в решении вопросов, связанных с отображением, обработкой и сбором картографических данных, наблюдаются следующие направления развития веб-картографии:

- перенос процедуры обработки картографических данных в веб;
- увеличение гибкости анализа пространственных данных.

Рассмотренные направления воплощаются в жизнь благодаря совершенствованию картографических инструментов, размещаемых на веб-серверах. Перенос обрабатывания и работы с картографическими данными в веб-среде явилось следствием непрерывного развития технологий в целом, что, в свою очередь, подтверждается увеличением интереса пользователей к новейшим инструментам работы с пространственными данными.

Литература

1. Публичная кадастровая карта [Электронный ресурс]. – Режим доступа: <http://maps.rosreestr.ru/PortalOnline/help.html#1> (дата обращения: 01.03.2017).
2. Быков А.В., Пьянков С.В. Web-картографирование: учеб. пособие. – Пермь: Перм. гос. нац. исслед. ун-т, 2015. – 110 с.
3. Зябликова Т.Л., Парфенов Ю.П. Веб-сервис для создания и публикации тематических карт // «Отечественная наука в эпоху изменений: постулаты прошлого и теории нового времени». Труды XI международной научно-практической конференции. – 2015. – № 6(11). – Ч. 2. – С. 48–52.
4. Дубинин М.Ю., Костикова А.М. Веб-ГИС // Компьютерра. – 2008. – № 33. – С. 22–28.



Довженко Михаил Игоревич

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: michael.dovzhenko@gmail.com



Готская Ирина Борисовна

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, д.педагог.н., профессор

e-mail: iringot@mail.ru

УДК 004

**АНАЛИЗ СПОСОБОВ РЕАЛИЗАЦИИ АЛГОРИТМА ОТСЛЕЖИВАНИЯ
ИЗМЕНЕНИЙ В ОДНОСТРАНИЧНЫХ ВЕБ-ПРИЛОЖЕНИЯХ**

Довженко М.И.

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 проанализированы различные подходы к отслеживанию изменений состояния одностраничных веб-приложений.

Ключевые слова: JavaScript, SPA, change detection, DOM, framework, рендеринг.

Главной задачей любого клиентского веб-приложения является отображение его внутреннего состояния на нечто, видимое на мониторе. Управление синхронизацией состояния приложения и пользовательского интерфейса уже давно является основным источником сложности в разработке пользовательского интерфейса, и к настоящему времени у нас есть несколько разных подходов к решению этого вопроса.

Разработчики берут набор объектов, массивов, строк и чисел и отображают их в дерево текстовых абзацев, форм, ссылок, кнопок и изображений.

Это само по себе достаточно просто: хотя отображение из модели данных в пользовательский интерфейс не всегда может быть тривиальным, это, в основном, по-прежнему просто функция с линейной логикой и предсказуемыми входными и выходными значениями. Все становится немного сложнее, когда мы начинаем говорить об изменениях данных во времени. Это может произойти, когда пользователь взаимодействует с пользовательским интерфейсом или, когда что-то еще происходит в мире, что приводит к обновлению данных. Пользовательский интерфейс должен отражать это изменение. Кроме того, поскольку изменение DOM-дерева является дорогостоящей операцией, хотелось бы отображать обновленные данные на экране при помощи минимального количества взаимодействий с DOM. Это гораздо более сложная проблема, чем просто одноразовый рендеринг пользовательского интерфейса, поскольку оно связано с изменениями состояния. Это та проблема, решения которой в различных фреймворках принципиально различаются.

1. Серверный рендеринг. До эпохи JavaScript каждое взаимодействие, которое пользователь осуществлял с веб-приложением, влекло за собой отправку запроса на сервер. Каждое нажатие и каждая отправка формы означали, что веб-страница закончила свою работу, запрос был отправлен на сервер, сервер ответил новой страницей, которую затем отобразил браузер. При таком подходе на front-end не было никакого состояния, чтобы управлять им. Какое бы ни было состояние, это было заботой серверной стороны. Клиентская часть была всего лишь сгенерированной HTML и CSS, с возможно небольшим количеством JavaScript. Хотя это был очень простой подход с точки зрения front-end, он также был очень медленным. Мало того, что каждое взаимодействие означало полный повторный рендеринг пользовательского интерфейса, это был также удаленный рендеринг с обращением к далекому дата-центру.

Большинство современных веб-приложений больше этого не делают. Мы можем отобразить начальное состояние нашей серверной части приложения, а затем переключиться на управление пользовательским интерфейсом на клиентской части.

2. Первое поколение JavaScript-приложений: ручное обновление UI. Первые поколения фреймворков JavaScript, такие как Backbone.js, ExtJS и Dojo, впервые представили реальную модель данных в браузере, вместо того, чтобы просто иметь легкие абстракции над взаимодействием с DOM. Это также означало, что приложения впервые изменили состояние в браузере. Содержимое модели данных может измениться, и тогда нужно было отобразить эти изменения на пользовательском интерфейсе.

Хотя эти фреймворки предоставили архитектуру для отделения кода пользовательского интерфейса от моделей, они все равно оставили синхронизацию между ними на разработчиков. Мы можем получать какие-то события, когда происходят изменения, но мы несем ответственность за то, что нужно обновить на UI и как это сделать.

Поддержка производительности такой модели также оставлена на разработчика. Поскольку разработчик получает контроль над тем, что и когда изменяется, он может настроить обновление на свое усмотрение. На этом месте часто возникает дилемма между простотой повторного рендеринга больших разделов страницы и производительностью повторного рендеринга только той части UI, которая нуждается в обновлении.

3. Ember: привязка данных и аксессоры. Необходимость вручную обновлять пользовательский интерфейс, когда состояние приложения изменяется, является одним из основных источников сложности в JavaScript-приложениях первого поколения. Ряд фреймворков направлен на устранение этой конкретной проблемы. Ember.js является одним из них.

Ember и Backbone отправляют события из модели данных, когда происходят изменения. Разница в том, что с Ember существует еще кое-что, что фреймворк обеспечивает для принимающего конца события. Он позволяет привязать пользовательский интерфейс к модели данных, а это значит, что есть обработчик для событий изменения, подключенный к пользовательскому интерфейсу. Этот слушатель знает, какие UI обновления делать, когда он получает событие.

Это приводит к довольно эффективному механизму обновления: хотя установка всех привязок сначала требует немного работы, после этого усилия, необходимые для синхронизации вещей, минимальны. Когда что-то меняется, активируются только те части приложения, которые действительно должны что-то сделать.

Большой компромисс такого подхода заключается в том, что Ember всегда должен знать об изменениях, которые происходят в модели данных [1]. Это означает, что данные модели нужно оборачивать в специальные объекты – аксессоры, унаследованные от API Ember'a, и что любые изменения над данными нужно выполнять с помощью специальных методов-сеттеров.

4. Грязная проверка с AngularJS. Как и Ember, AngularJS также стремится решить проблему ручного повторного рендеринга, когда все изменилось. Однако он подходит к проблеме

под другим углом. Когда в шаблоне Angular указана ссылка на свои данные, например, в выражении типа `{{foo.x}}`, AngularJS не только отображает эти данные, но также создает наблюдателя для этого конкретного значения. После этого, когда что-либо происходит в приложении, Angular проверяет, изменилось ли значение в этом наблюдателе с последнего раза. Если изменилось, он обновляет значение в пользовательском интерфейсе. Процесс запуска этих наблюдателей называется грязной проверкой.

Огромное преимущество этого стиля обнаружения изменений заключается в том, что он позволяет использовать любые структуры данных в моделях. Не нужно наследоваться ни от каких базовых классов и реализовывать какие-либо интерфейсы.

Недостатком является то, что, поскольку в модели данных нет никаких механизмов, которые бы говорили фреймворку об изменениях, фреймворк не имеет никакого представления о том, где и где они могут произойти. Это означает, что модель должна быть проверена на предмет изменений извне, и это именно то, что делает Angular.

Запуск всех наблюдателей каждый раз может показаться неэффективным, но это может быть быстро. Это происходит главным образом из-за того, что обычно не происходит работа с DOM до тех пор, пока изменения не будут обнаружены, а простые проверки объектных ссылок на равенство в JavaScript являются дешевыми. Но при наличии комплексных пользовательских интерфейсов или при необходимости обновлять данные слишком часто, может потребоваться дополнительная оптимизация производительности [2].

5. React: виртуальный DOM. React, как Angular, не навязывает свой интерфейс для модели данных и позволяет использовать любые объекты и структуры данных, которые лучше подходят. Однако то, что делает React, действительно возвращает нас к старым добрым временам рендеринга на стороне сервера, когда нас просто не волновали изменения состояния: он обновляет весь пользовательский интерфейс с нуля каждый раз, когда в нем происходит какое-то изменение. Это может значительно упростить код пользовательского интерфейса. Разработчику не нужно заботиться о поддержке состояния в компонентах React. Как и при рендеринге на стороне сервера, мы рендерим один раз и все готово. Когда происходит изменение в компоненте, он автоматически рендерится повторно. Нет никакой разницы между первоначальным рендерингом компонента и его обновлением для измененных данных.

Это звучит очень неэффективно, и было бы, если бы это был конец истории. Однако повторный рендеринг у React происходит особым способом. Когда рендерится пользовательский интерфейс, React сначала создает виртуальный DOM, который не является фактическим графом объектов DOM, а представляет собой легкую, чистую структуру данных JavaScript из простых объектов и массивов, которая эмулирует реальный граф объектов DOM. Затем отдельный процесс берет эту виртуальную структуру DOM и создает соответствующие реальные элементы DOM, которые отображаются на экране. Затем, когда происходит изменение, создается новый виртуальный DOM с нуля. Этот новый виртуальный DOM будет отражать новое состояние модели данных. Теперь у React есть две виртуальные структуры данных DOM: новая и старая. Затем он запускает специальный алгоритм сравнения двух виртуальных DOM, чтобы получить все изменения между ними. Эти и только эти изменения применяются к реальному DOM: этот элемент был добавлен, значение этого атрибута изменено и т.д.

6. React: неизменяемые структуры данных. Хотя подход React с виртуальным DOM довольно быстрый, он все еще может стать узким местом, когда UI состоит из большого количества компонентов или, когда его нужно обновлять очень часто (скажем, до 60 раз в секунду). Проблема в том, что при изменении состояния некоторого компонента React создает с нуля виртуальное DOM-дерево этого компонента с нуля, если не ввести какой-либо контроль над способом изменения модели данных.

Один из подходов к контролю изменений заключается в том, чтобы использовать неизменяемые, постоянные структуры данных. Особенность неизменяемых структур данных

состоит в том, что, как следует из названия, их нельзя мутировать, а только создавать новые версии. Если нужно изменить свойство объекта, нужно создать новое свойство с новым атрибутом, так как нельзя изменить существующий. Благодаря тому как работают неизменяемые структуры данных, это на самом деле намного эффективнее, чем кажется. С точки зрения обнаружения изменений это означает то, что, когда состояние компонента React состоит только из неизменяемых данных и во время рендеринга состояние компонента указывает на ту же структуру данных, что и во время последнего рендеринга, разработчик может сказать фреймворку пропустить повторный рендеринг. React может просто использовать предыдущий виртуальный DOM для этого компонента и всего дерева компонентов, вытекающих из него. Нет необходимости проверять дальше, поскольку ничего не могло измениться в состоянии компонента. Производительность рендеринга повышается многократно [3].

Обнаружение изменений является центральной проблемой в разработке пользовательского интерфейса, а JavaScript фреймворки пытаются решить его различными способами.

Ember обнаруживает изменения, когда они происходят, поскольку он управляет API-интерфейсом модели данных и может генерировать события, когда вызываются его методы.

AngularJS обнаруживает изменения постфактум, повторно опрашивая всех зарегистрированных наблюдателей, чтобы узнать, изменились ли их значения.

React обнаруживает изменения путем создания виртуального DOM-дерева на основе пользовательского интерфейса, а затем сравнивает его со старой версией. Все, что изменилось, попадает в реальный DOM.

Неизменяемые структуры данных повышают производительность рендеринга React, позволяя пометить DOM-дерево компонента как неизменное, потому что мутации в состоянии компонента не допускаются [4].

Литература

1. Change detection overview explained [Электронный ресурс]. – Режим доступа: <https://gofore.com/en/change-detection-overview-part-1/>, своб.
2. Аксессоры против грязной проверки в JavaScript фреймворках [Электронный ресурс]. – Режим доступа: <http://blog.bguiz.com/2013/08/05/accessors-vs-dirty-checking-in-javascript-frameworks/>, своб.
3. Angular change detection explained [Электронный ресурс]. – Режим доступа: <https://blog.thoughttram.io/angular/2016/02/22/angular-2-change-detection-explained.html#smarter-change-detection>, своб.
4. Change and its detection in JavaScript frameworks [Электронный ресурс]. – Режим доступа: <https://teropa.info/blog/2015/03/02/change-and-its-detection-in-javascript-frameworks.html>, своб.



Дорофеева Мария Игоревна

Год рождения: 1996

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4106

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: marie.dorofeeva@yahoo.com



Погорелов Виктор Иванович

Год рождения: 1940

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, д.т.н., профессор

e-mail: vic@VP2098.spb.edu

УДК 004.51

МЕТОДЫ ОЦЕНКИ ЮЗАБИЛИТИ ВЕБ-ИНТЕРФЕЙСОВ

Дорофеева М.И., Погорелов В.И.

Научный руководитель – д.т.н., профессор Погорелов В.И.

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

В работе рассмотрено современное состояние области человеко-ориентированного проектирования, проанализирована роль юзабилити в процессе создания веб-интерфейсов. Проведен обзор современных методов оценки юзабилити веб-сайтов с учетом их характеристик.

Ключевые слова: UX, юзабилити, пользовательский интерфейс, человеко-ориентированное проектирование, юзабилити-тестирование.

В настоящее время правильное с точки зрения юзабилити проектирование пользовательских интерфейсов – человеко-ориентированное – и своевременное проведение юзабилити-тестирования на реальных пользователях является залогом создания успешных по потребительским показателям цифровых продуктов, в том числе, веб-сайтов [1]. В зависимости от целей интернет-проекта и предоставляемых услуг, визуальный вид и внутренняя логика сайтов может значительно различаться, что подразумевает различные подходы к оценке юзабилити. Приведенные обстоятельства говорят о важности проработки требований к интерфейсу еще до этапа проектирования. В последние годы наблюдается тенденция смещения фокуса в сторону эстетических качеств интерфейса, эмоционального отклика пользователя в ущерб эргономическим характеристикам, что влияет на сферу юзабилити-тестирования.

Юзабилити как свойство цифрового продукта связано с возможностью использования и индивидуальной оценкой такого использования определенной группой пользователей. Можно утверждать, что для продукта не существует внутреннего юзабилити, есть только способность быть используемым предполагаемой группой пользователей в некоторых условиях – в определенном контексте. Из этого следует, что удобство работы с интерфейсом невозможно оценить изолировано, вне этого контекста. Именно с этой проблемой сталкиваются современные разработчики веб-интерфейсов, игнорируя важность тестирования на реальных пользователях, предпочитая этому исследования предполагаемой целевой аудитории (UX, user experience – опыт взаимодействия) и более доступные и менее затратные средства веб-аналитики или экспертной оценки [2].

Нельзя недооценивать исследования целевой аудитории и пользовательского опыта как важного этапа проектирования. Однако данные, полученные в результате качественных UX-исследований не заменяют собой тестирование продукта во время итеративного процесса разработки и детализации, поскольку всегда присутствующий человеческий фактор и особенности человеческой психологии не позволяют точно предсказать нюансы человеко-компьютерного взаимодействия. Без эмпирической составляющей, т.е. без проведения экспериментов в процессе человеко-ориентированного проектирования конечный результат может значительно расходиться с планируемым и, как следствие, не оправдывать ожиданий разработчиков и пользователей [3].

Обилие существующих методов исследования, как пользовательской аудитории, так и интерфейсов часто представляют дополнительную сложность для создателей [4], особенно в условиях ограниченного бюджета и сжатых сроков выполнения проекта. В зависимости от специфики решаемых задач и требуемых условий в рамках одного проекта необходимо использовать наиболее подходящий метод или комбинацию методов.

Укрупненная классификация методов исследования юзабилити выглядит следующим образом [5]:

1. методы с прямым участием пользователей (табл. 1);
 - методы, основанные на наблюдении за поведением пользователя;
 - методы, основанные на получении субъективной оценки пользователя;
2. методы с непрямым участием пользователей (табл. 2).

Таблица 1. Методы оценки юзабилити с прямым участием пользователей

Метод	Достоинства	Недостатки	Этап разработки
Лабораторное юзабилити-тестирование	– самая надежная информация о взаимодействии пользователя с интерфейсом	– высокая или средняя стоимость; – высокие временные затраты; – необходимо наличие профессионального опыта	Тестирование на прототипах на этапе детализации продукта и на финальной стадии
Немодерируемое юзабилити-тестирование	– меньшая стоимость по сравнению с предыдущим методом	– уменьшение точности и надежности	Тестирование на прототипах на этапе детализации продукта и на финальной стадии
Айтрекинг	– точные данные о взаимодействии с интерфейсом; – глубокий анализ дизайна	– высокая стоимость; – не заменяет другие методы оценки; – необходимо наличие профессионального опыта	Тестирование на прототипах на этапе детализации продукта и на финальной стадии
Веб-аналитика	– быстрая оценка; – невысокая стоимость; – большой охват аудитории	– средняя точность и надежность; – не выявляет причины проблем	Тестирование бета-версии или готового продукта
Интервью	– самые подробные качественные данные; – гибкость подхода	– проблемы толкования и анализа; – большие временные затраты	Тестирование бета-версии или готового продукта

Метод	Достоинства	Недостатки	Этап разработки
Анкетирование	– формализованный подход к измерению удовлетворенности	– вероятность ошибок в вопросах и ответах	Тестирование бета-версии или готового продукта
Опросы	– быстрое получение качественной оценки; – невысокая стоимость	– средняя точность и надежность; – вероятность ошибок в вопросах и ответах	Тестирование бета-версии или готового продукта

Таблица 2. Методы оценки юзабилити с непрямим участием пользователей

Метод	Достоинства	Недостатки	Этап разработки
Экспертная оценка	– невысокая стоимость;	– вероятность упущения важных проблем; – средние или высокие временные затраты	Тестирование бета-версии или готового продукта
Автоматическая оценка	– невысокая стоимость; – быстрая оценка	– вероятность упущения важных проблем; – экспериментальный способ	Тестирование бета-версии или готового продукта
Метод на основе моделей	– доступность и низкая стоимость; – улучшает понимание процесса взаимодействия на этапе прототипирования	– необходимо наличие профессионального опыта для создания персонажей и маршрутов; – средние и высокие временные затраты	Тестирование на прототипах на этапе детализации продукта и на финальной стадии

Например, не для каждого веб-интерфейса необходимо исследование с помощью устройств айтрекинга, в случае если создается интернет-магазин или приложение, аналогичные по структуре существующим на рынке. Технология айтрекинга дорогостоящая, а получаемые результаты достаточно сложны в обработке и анализе. Применение метода оценки юзабилити с использованием айтрекинга оправдано в случае создания радикально новых визуальных приемов и способов взаимодействия с пользователем или в случае, когда проведение более доступных исследований не выявило причин отрицательного опыта взаимодействия и проблем с юзабилити.

Наиболее подходящий метод оценки потребительских качеств любого интерфейса – юзабилити-тестирование. Тестирование на реальных пользователях способно определить причины возникновения проблем во взаимодействии человека с интерфейсом, в отличие от других способов, которые определяют наличие и качество проблем. Это позволяет на ранних этапах разработки веб-сайта или веб-приложения выявить недостатки и ошибки, которые могут привести к проблемам в дальнейшем, и устранить их с минимальными потерями для проекта.

В дальнейшем исследовании планируется рассмотреть особенности восприятия человеком информации в веб и возможности окулографических исследований.

Литература

- ГОСТ Р ИСО 9241-210-2016. Эргономика взаимодействия человек-система. Часть 210. Человеко-ориентированное проектирование интерактивных систем. – Введен 01.12.2017. – М.: Стандартинформ, 2016. – 32 с.

2. Бурмистров И.В. Кризис в юзабилити-инженерии и отсутствие предпосылок к его преодолению // Эрго 2016: человеческий фактор в сложных технических системах и средах. Межрегиональная эргономическая ассоциация – 2016. – С. 321–328.
3. Nielsen J. First Rule of Usability? Don't Listen to Users [Электронный ресурс]. – Режим доступа: <https://www.nngroup.com/articles/first-rule-of-usability-dont-listen-to-users/>, своб.
4. Rohner Ch. When to Use Which User-Experience Research Methods [Электронный ресурс]. – Режим доступа: <https://www.nngroup.com/articles/which-ux-research-methods/>, своб.
5. Батенькина О.В. Методы оценки удовлетворенности пользователей при тестировании юзабилити информационных систем // Визуальная культура: дизайн, реклама, информационные технологии. Сб. трудов XV международной научно-практической конференции. Омский государственный технический университет. – 2016. – С. 151–154.



Дятлов Егор Александрович

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4108

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: e.d.webprogrammer@gmail.com



Шалобаев Евгений Васильевич

Год рождения: 1947

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.т.н., доцент

e-mail: kpd@limtu.ru

УДК 004.421.4

МОДЕЛИ АНАЛИЗА ПОИСКОВОЙ ОПТИМИЗАЦИИ ВЕБ-РЕСУРСА

Дятлов Е.А.

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 описаны этапы работы поисковых систем российского Интернет: Google и Яндекс; а также представлен обзор и анализ моделей анализа поисковой оптимизации веб-ресурса.

Ключевые слова: поисковая система, поисковая оптимизация, релевантность веб-ресурса, ранжирование, факторы ранжирования, функция ранжирования, модели анализа.

Развитие глобальной сети Интернет и увеличение количества веб-ресурсов, размещенных в ней, актуализировали проблемы поиска информации и конкуренции между выдаваемыми веб-документами по соответствующим (релевантным) запросам пользователя.

Проблему поиска информации решают алгоритмы поисковых систем. Их задача заключается в том, чтобы расположить найденные веб-документы в порядке убывания релевантности. Процесс упорядочивания документов называется ранжированием [1].

Работа поисковой системы состоит из трех частей:

1. сбор данных;
2. индексирование;
3. поиск.

Сбор данных начинается с того, что поисковый робот (crawler) непрерывно формирует каталог адресов веб-документов. Он просматривает все документы из каталога и собирает с них гипертекстовые ссылки, содержащие адреса на другие веб-ресурсы, а затем полученные адреса добавляет в каталог, которому был задан начальный набор адресов при разработке поисковой системы.

Индексирование – один из самых сложных процессов работы поисковой системы. В его основе лежат алгоритмы машинного обучения и нейронные сети, которые анализируют информацию о веб-документе, полученную на этапе сбора данных [1]. Тексты, свойства, характеристики веб-ресурса разбиваются на части и анализируются поисковым роботом. Затем полученные данные заносятся в базы данных под соответствующим индексом для

быстрого поиска, и со ссылкой на соответствующий веб-документ [2]. Такие данные называются факторами ранжирования.

В отличие от сбора данных и индексирования, процесс работы поиска частично виден пользователю и требует его взаимодействия [2].

Для того чтобы определить релевантность веб-документа и упорядочить его в результатах поиска, поисковому роботу необходимо учитывать не только факторы веб-ресурса, но и факторы запроса пользователя. Например, географического местоположение пользователя и морфологию запроса.

Каждый из факторов имеет свой собственный вес в функции ранжирования. Она упорядочивает веб-документы в порядке убывания релевантности по соответствующему запросу пользователя [1]. Факторы ранжирования имеют отношения между собой. Добавление или изменение фактора в функции ранжирования может усилить или ослабить другие [3].

Результат функции ранжирования – это итог работы искусственного интеллекта. Именно он определяет вес факторов ранжирования и связь между ними. Формально процесс вычисления релевантности веб-документа представлен на рис. 1.



Рис. 1. Процесс вычисления релевантности

Чтобы поисковые роботы смогли найти веб-документ и максимально точно его проиндексировать, необходимо произвести поисковую оптимизацию [4]. Поисковая оптимизация – это комплекс мер по анализу процессов работы поисковых систем, их алгоритмов, с целью повышения позиций веб-документа в результатах поиска по соответствующим релевантным запросам пользователя [1].

Для успешного продвижения веб-ресурса в поисковых системах необходимо понимать, какие факторы ранжирования стоит учитывать, а также их степень значимости.

Задача модели анализа поисковой оптимизации заключается в том, чтобы определить: могут ли параметры, которые исследователь приписывает к веб-ресурсу, быть факторами ранжирования и насколько сильными.

В 2010 году было проведено масштабное исследование факторов ранжирования [5]. При этом была выбрана модель анализа, основанная на методе экспертных оценок. В исследовании приняло участие 26 экспертов: руководители крупнейших компаний, специалисты, успешно продвигающие сайты в самых конкурентных тематиках, разработчики программ и сервисов, которые известны на рынке поисковой оптимизации.

В исследовании рассматривалось более 200 факторов, и каждый из них оценивался экспертом по критерию влияния на ранжирование:

0 – не влияет;

1 – влияет незначительно;

- 2 – стоит учитывать;
- 3 – сильно влияет;
- 4 – важнейший фактор.

Оценки каждого фактора суммировались, после чего было подсчитано процентное соотношение значимости каждого фактора от значимости того, который имеет самый высокий рейтинг. Все факторы были отсортированы по степени важности и отображены в процентных долях.

Итоги исследования стоит воспринимать следующим образом:

- 0–20% – фактор не влияет на релевантность;
- 20–40% – влияние фактора незначительно;
- 40–60% – фактор важно учитывать при поисковой оптимизации;
- 60–80% – фактор значительно влияет на релевантность;
- 80–100% – фактор необходимо учитывать обязательно.

Недостатки этого исследования заключаются в его субъективности, а также достоверность и надежность результатов зависят от компетентности экспертов.

Более точные результаты показывает модель анализа поисковой оптимизации, основанная на методах математической статистики или машинного обучения, но для этого требуются большое количество данных и ресурсов [6]. Такие исследования могут позволить себе крупные компании, занимающиеся разработкой программного обеспечения, связанного с продвижением веб-ресурсов, и заинтересованные в этом с коммерческой стороны.

Подобные исследования проводили как зарубежные, так и отечественные компании: Searchmetrics, SEMrush, «Ашманов и партнеры». Результаты были опубликованы на соответствующих официальных сайтах компаний в 2017 году. Обзор и анализ исследований представлен на рис. 2.

	Searchmetrics	SEMrush	«Ашманов и партнеры»
Количество параметров	менее 50	17	около 500
Страна	США	США и др.	Россия
Факторы	ссылочные, текстовые, технические, социальные, поведенческие	поведенческие, ссылочные, текстовые, технические	коммерческие, ссылочные, текстовые, технические, социальные
Поисковые машины	Google	Google	Яндекс, Google, Mail.ru
Наличие фона	нет	нет	да
Оценка попадания в топ-30	нет	нет	да
Количество запросов	не указано	600 000	от 160 до 10 000 в разных исследованиях
Отбор запросов	частотная верхушка: общая выборка, e-commerce, туризм, финансы, медицина и др.	разделены на 4 группы по частоте	сбалансированы по частоте и тематикам
Типы запросов	информационные, навигационные, коммерческие	информационные, навигационные, коммерческие	коммерческие
Оценки ассессоров	нет	нет	да
Методы	статистические: ранговый критерий Спирмена	деревья решений (random forest)	статистические: ранговый критерий Спирмена, точный тест Фишера, U-критерий Манна-Уитни и др.

Рис. 2. Обзор и анализ исследований компаний Searchmetrics, SEMrush, «Ашманов и партнеры»

Исследование компании «Ашманов и партнеры» является более содержательным, поскольку в нем оценивалось около 500 факторов ранжирования, рассматривались

поисковые системы: Яндекс, Google, Mail.ru; а также применялась совокупность методов математической статистики, что подразумевает более достоверные результаты.

Исследование компании «Ашманов и партнеры» проводилось на неизменной выборке из 160 коммерческих запросов и заключалось в получении ответов на следующие вопросы:

1. Какие параметры статистики достоверно коррелируют с позицией в результатах поиска?
2. Какие параметры статистически достоверно связаны с попаданием в топ-3, в топ-10?
3. Какие параметры статистически достоверно связаны с попаданием в топ-30?

При помощи методов математической статистики можно достаточно уверенно ответить на все три вопроса.

Для оценки корреляции параметра с позицией в результатах поиска в первую очередь используется ранговый критерий Спирмена, а для оценки связи с попаданием в топ-30 – точный тест Фишера. Коэффициент корреляции Пирсона показывает, похожа ли зависимость между позицией в выдаче и некоторым признаком на линейную (чем дальше от 0, тем больше похожа). А для того, чтобы определить статистическую значимость фактора (лучшее ранжирование), применяется статистический тест U-тест Манна–Уитни, который сравнивает между собой позиции сайтов с признаком и без признака, или со значением признака выше/ниже медианного [6].

Модель анализа поисковой оптимизации веб-документа, основанная на методах математической статистики, формально представлена на рис. 3.

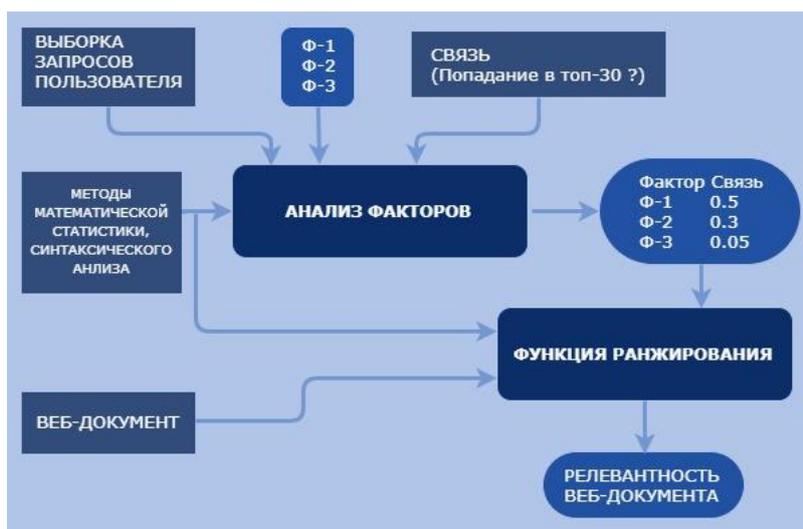


Рис. 3. Модель анализа поисковой оптимизации веб-документа

Литература

1. Ашманов И.С., Иванов А.А. Оптимизация и продвижение сайтов в поисковых системах. – 3-е изд. – СПб.: Питер, 2011. – 464 с.
2. Гроховский Л., Сливинский М., Чекушин А., Ставский С. SEO: руководство по внутренним факторам. – М.: Центр исследований и образования «ТопЭксперт.РФ», 2011. – 133 с.
3. Неелова Н. Энциклопедия поискового продвижения Ingate. – Изд. 2-е. – ИП Андросов, 2017. – 541 с.
4. Лысенко Д.С., Гридина Е.Г. Факторы формул ранжирования поисковых систем Яндекс и Google // Журнал «Качество. Инновации. Образование». – 2011. – № 3. – С. 35–41.
5. SEO-альманах 2010 [Электронный ресурс]. – Режим доступа: <http://www.topexpert.pro/1.html> (дата обращения: 23.10.2017).
6. Яндекс и Google факторы ранжирования в 2017 году [Электронный ресурс]. – Режим доступа: https://www.ashmanov.com/static/aip_lab_otchet.pdf (дата обращения: 17.12.2017).



Захарова Анастасия Олеговна

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4206

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: aozakharova18@gmail.com

УДК 004.92

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СРЕДСТВ ВИЗУАЛИЗАЦИИ ДАННЫХ

Захарова А.О.

Научный руководитель – д.т.н., профессор Погорелов В.И.

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

В работе определено понятие «визуализации данных», приведена классификация типов визуализации данных по категориям, представлены результаты сравнительного анализа средств визуализации данных.

Ключевые слова: визуализация, инфографика, средства визуализации, типы визуализации, принципы визуализации, программные средства.

Задача любой визуализации – представить информацию упрощенно, позволив зрителю увидеть конечный результат вычислений. При этом идеальная визуализация является понятной сама по себе, и может сохранить свой смысл, даже лишившись всего сопровождающего текста. Визуализация данных – инструмент исследования и анализа больших числовых массивов, относительно неупорядоченных, необработанных, позволяющий изучать отдельные показатели и общие закономерности [1–4].

Спецификой «визуализации данных» является то, что она представляет из себя визуализированный массив числовых и категориальных данных. Причем в материале отсутствует связанный и цельный текст.

На сегодняшний день различают 138 типов визуализации данных.

По типу их можно разделить на следующие категории:

1. семейство исходных данных:
 - диаграммы;
 - схемы;
 - геопространственные данные;
 - графики;
 - таблицы;
2. функция:
 - сравнение;
 - концептуальная визуализация;
 - корреляция;
 - распределение;
 - географические данные;
 - временная динамика;
 - часть от целого;
3. геометрические формы:
 - площадь;
 - планка;

- круг;
- точки;
- иконки;
- линии;
- карта;
- полигоны (графы);
- пирамида;
- квадрат.

В этой работе не были рассмотрены примеры качественной и некачественной визуализации данных, однако, исходя из накопленного опыта, можно сделать вывод, что есть три общих правила, которых следует придерживаться при создании качественной визуализации данных:

- визуализация должна достаточно емко отвечать на поставленный вопрос;
- визуализация должна быть понятна без сопроводительного текста;
- в визуализации не должно быть избыточной информации.

Избыточность информации можно проверить следующим способом – удалить информацию и проанализировать результат, если информативности визуализации не уменьшилась, то лишнюю информацию необходимо убрать. В наше время, когда вокруг столько данных навык правильного извлечения, обработки и представления информации – залог успеха любого профессионала.

Что же касается выбора специализированных средств для создания инфографики, то выбор опирается на достижение поставленных целей. Наличие большого количества предлагаемого программного обеспечения привело к тому, что его также следует классифицировать по типу создаваемой инфографики.

По типу программного обеспечения, средства создания инфографики делятся на:

- создание диаграмм, графиков, блок-схем;
- создание таймлайнов;
- создание иконок;
- создание карт;
- создание видеоинфографики;
- создание интеллектуальных карт;
- создание резюме;
- создание презентаций;
- создание облака слов;
- платформы, содержащие исходный код;
- создание фотоинфографики.

Общеизвестно, что, до того, как возникла потребность в качественном представлении информации, при создании графиков использовался продукт компании Microsoft – Microsoft Excel. С течением времени и развитием технологий для информативного изображения тех же диаграмм стали использовать другие более сложные программы. На сегодняшний день существует огромное количество различных приложений, онлайн-сервисов и программного обеспечения для создания инфографики.

Классификация программного обеспечения по типу создаваемых данных:

1. векторная графика:
 - Adobe Illustrator;
 - продукты Corel Draw, Designer, Painter;
 - Microsoft Expression Studio;
 - Inkscape;
2. растровая графика:
 - Adobe Photoshop;
 - Corel Photo Painter;

- Gimp;
- 3. 3D-графика:
 - AutoDesk;
 - 3D MAX;
 - AutoDesk 3D Maya;
 - Maxon Cinema 4d;
- 4. анимация и интерактивность:
 - Adobe Flash;
 - Adobe Flash Builder;
 - технологии XML и PHP;
- 5. видеоконтент:
 - Adobe Premiere;
 - Adobe After Effects.

Универсальные графические редакторы, например, продукт компании Adobe – Adobe Illustrator, не всегда лучший выбор. С одной стороны, они позволяют создать действительно уникальную инфографику, с другой – требуют хорошей подготовки и перегружены ненужными инструментами.

Зачастую в ситуации, когда нужно быстро и просто визуализировать информацию, не прибегая к сложным инструментам, выгоднее использовать специализированные редакторы, так как они содержат только необходимые для решения поставленной задачи инструменты, а также ориентированы на специфические запросы именно такого типа контента. Кроме того, они предлагают свои инструменты импорта больших объемов «сырых» данных, а иногда и доступ к уже готовым базам данных.

В таблице представлены примеры таких редакторов.

Таблица. Примеры редакторов

Название сервиса	Описание
Piktochart	Веб-инструмент, содержащий шаблоны для создания простой графики. Здесь можно добавлять простые графики, гистограммы и секторные диаграммы на основе данных из CSV-файлов или вводить их напрямую. Также предусмотрен экспорт файлов в форматы PNG и JPG в разрешениях для печати и для веб.
Easel.ly	Подходит для визуализации идей и историй. Инструмент отличается самым широким набором объектов (люди, коллекции иконок, достопримечательности, карты, животные и т.д.) и фонов. JPG-файл можно сохранить в разрешении для веб.
Infogr.am	Обладает прекрасным интерфейсом для создания простой инфографики и имеет в своем распоряжении несколько красивых шаблонов. Есть возможность загружать собственные изображения или видео.
Visual.ly	Данный сервис предлагает несколько простых инструментов, многие из которых интегрируются с социальными сетями и анализируют данные Twitter и Facebook. На данном сайте была создана специальная коммерческая площадка для сотрудничества с дизайнерами-визуализаторами и другими специалистами, специализирующимися на инфографике.
Tableau	Не является веб-приложением, но его можно установить на свой компьютер. С его помощью можно загружать таблицы и данные из любых CSV-файлов и создавать различные интерактивные визуализации, включая тепловые карты, отражающие активность действий в разных областях поля, диаграммы Венна для представления связей, столбчатые диаграммы, графики и другие схемы.

Литература

1. Пескова О.В. О визуализации информации // Вестник МГТУ им. Н.Э. Баумана. – Сер. «Приборостроение». – 2012. – С. 1.
2. Силанов Н.А. Информационная графика в современной визуальной культуре. // Вестник Московского университета. Серия 10. Журналистика. – М.: Изд-во МГУ. – 2010. – № 3. – С. 29.
3. Доброва И. Плотность данных [Электронный ресурс]. – Режим доступа: <http://infographer.ru/plotnost-dannyx/>, своб.
4. Лима М. Манифест визуализации информации [Электронный ресурс]. – Режим доступа: http://www.vmethods.ru/2010/05/blog-post_18.html/, своб.



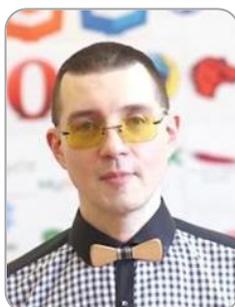
Здоренко Алиса Алексеевна

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4108

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: zdorenko.alisa@gmail.com



Государев Илья Борисович

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: goss@corp.ifmo.ru

УДК 004.77

ПРОЕКТИРОВАНИЕ И РАЗВИТИЕ ФРЕЙМВОРКОВ НА ПЛАТФОРМЕ NODE.JS

Здоренко А.А.

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 объектом исследования являлись фреймворки на платформе Node.js. Проведен сравнительный анализ существующих наиболее популярных фреймворков на платформе Node.js.

Ключевые слова: платформа Node.js, фреймворк, веб-приложение, сайт.

В современном мире скорость разработки сайтов и веб-приложений имеет большое значение, так как с каждым днем растет спрос на данные виды услуг, что требует в короткие сроки выполнять поставленные задачи. Вопрос разработки веб-приложений с помощью фреймворка на платформе Node.js в последние годы оказывается в фокусе исследовательского внимания российских и зарубежных исследователей, однако полностью проблематика данного вопроса в современных отечественных и частично зарубежных исследованиях раскрыта недостаточно детально. Сегодня проблема выбора правильного фреймворка является одной из самых актуальных, потому что от данного выбора зависит насколько быстро и эффективно будет написан проект. Некоторые фреймворки предназначены для больших проектов, которые предполагают доработки проекта под нужды конкретного потребителя и долгосрочную поддержку приложения. Другие подходят для быстрого старта проекта, которые не предполагают расширения в будущем и долгосрочной поддержки [1–3].

На сегодняшний день существует достаточно много фреймворков на платформе Node.js для создания веб-приложений и сайтов. Из-за большого количества фреймворков невозможно определиться с выбором. Главное различие между фреймворками – это структура проекта, базовый функционал, маршрутизация, хранение данных. В данной работе рассмотрены достоинства и недостатки четырех популярных и современных фреймворков на платформе Node.js:

- фреймворк Express был выпущен в 2010 году, по сравнению с остальными фреймворками он является достаточно «старым» и благодаря своей стабильности прочно занимает позицию одного из самых популярных фреймворков. Он быстр, компактен, не навязывает разработчику жестких архитектурных решений. В основе его стремительного развития лежит простота и понятность;

- Koa.js – фреймворк, созданный той же командой, которая занимается Express, создатели стремились учесть все недостатки предшественника и сделать его более современным и удобным в использовании. Его продвигают как «фреймворк следующего поколения для Node.js». Его можно охарактеризовать как систему, которая отличается компактностью, выразительностью и надежностью;
- Sails.js – это MVC-фреймворк, который позволяет легко и быстро создавать Node.js-приложения. Этот фреймворк особенно хорошо подходит для разработки чатов, инструментальных панелей реального времени и многопользовательских игр. Фреймворк Sails разработан как полный готовый продукт, который уже включает в себя достаточный функционал для того, чтобы можно было начать работу, и при этом использует минимальное количество внешних модулей. В результате чего он позволяет в короткие сроки развернуть готовое приложение с необходимым функционалом;
- Meteor – один из самых популярных JS-фреймворков с большим количеством функций для back-end разработки и отрисовки front-end, управления базами данных и бизнес-логикой. Первый релиз был в 2012 году, с этого момента его экосистема выросла кардинально быстрыми темпами. Эта fullstack-платформа позволяет быстро разрабатывать веб и мобильные приложения. Однако использование Meteor может усложниться, если нужно что-то, что не встроено в систему.

В процессе исследования фреймворков было выявлено, что разработка веб-приложений с помощью фреймворка на платформе Node.js является важной частью для достижения высокой эффективности и разработки проектов. Главное правильно выбрать фреймворк. Выбор фреймворка зависит от первоначальных целей компании, требований к проекту, общей функциональности фреймворка и как ее можно применить в каждом конкретном случае. Исходя из этого, общий вывод был сделан по каждому рассмотренному фреймворку.

Express-фреймворк не ограничивает разработчика в использовании встроенных модулей, а дает возможность выбрать из множества именно тот, который подходит лучше всего для конкретного проекта. Обычно на нем пишутся большие проекты, которые предполагают кастомизацию и когда необходима долгосрочная поддержка приложения.

Koa.js подходит для любых проектов, которые предполагают развитие в перспективе. Koa вобрал в себя достоинства проверенного и широко-используемого фреймворка Express, были учтены недостатки. Koa подходит для разработки самых разнообразных приложений любого объема, с любой степенью кастомизации и с любыми требованиями к поддержке.

Sails.js подходит для быстрого старта проекта, которые не предполагают расширения в будущем и долгосрочной поддержки. Этот фреймворк подходит для начинающих Node.js программистов.

Meteor.js подходит для быстрой и простой разработки. В нем довольно много новых и интересных идей. Meteor представляет собой fullstack-платформу, благодаря чему можно повторно использовать код на сервере и клиенте.

Исследование показало, что можно найти информацию в отдельности, про каждый фреймворк, но проблема выбора подходящего фреймворка на платформе Node.js не раскрыта полностью в существующих трудах, как правило, авторы останавливаются на одном фреймворке, описывая его функционал. А материала по сравнительному анализу фреймворков недостаточно, и зачастую информация является неактуальной.

Литература

1. Witten I.H., Frank E., Hall M.A. Data Mining: Practical Machine Learning Tools and Techniques. – Morgan Kaufmann Publishers Inc. San Francisco. – 2011. – 630 p.
2. Isoni A. Machine Learning for the Web. – Packt Publishing, 2016. – 298 p.
3. Raschka S., Mirjalili V. Python Machine Learning. – 2nd Edition. – Packt Publishing, 2017. – 622 p.



Зиннатулин Фаиль Фидаэлевич

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4106

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: prosomf@gmail.com



Шалобаев Евгений Васильевич

Год рождения: 1947

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.т.н., доцент

e-mail: kpd@limtu.ru

УДК 004.921

АНАЛИЗ ИССЛЕДОВАНИЙ ТРЕХМЕРНОГО МОДЕЛИРОВАНИЯ В МЕДИЦИНЕ

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 рассмотрены направления применения трехмерного моделирования в медицине; обобщены отечественные и зарубежные исследования проблемы трехмерного моделирования в медицине. Выделены методы обработки поверхности созданных трехмерных моделей и способы автоматизации создания трехмерных моделей.

Ключевые слова: трехмерное моделирование, анатомия, биопечать, томография, диагностика.

Проникновение информационных технологий в различные области науки во многом определяет прогресс и направления развития в этих областях [1]. Медицина – это чрезвычайно сложная область исследований, которая, как и любая другая дисциплина, имеет фундаментальное значение в человеческом существовании [2]. Модели были использованы для изучения анатомии человека на протяжении более тысячи лет. При изготовлении моделей использовались такие материалы как воск, бронза. В последнее время компьютерные технологии начали вытеснять традиционный способ обучения визуальной насыщенной и зачастую интерактивной трехмерной (3D) компьютерной графикой. Возможность вращать трехмерное изображение черепа и видеть его внутреннее строение способствовала наилучшему восприятию взаимосвязей, что позволило хирургам наиболее точно спланировать и прогнозировать результат оперативного вмешательства [3]. 3D-моделирование также способствует эффективному донесению информации до аудитории, которая включает как работников в медицине, так и общественность в целом, наглядно демонстрируя научные концепции. 3D-модели могут быть использованы в качестве элементов систем хирургического обучения в виртуальной реальности. Благодаря этому хирургические процедуры могут преподаваться хирургами во время проведения операций, позволяя хирургам-стажерам получать опыт без риска для жизни пациентов с помощью интерфейсов на тактильной основе, которые содержат датчики дополнительного осязания, которые позволяют почувствовать пользователям силу реального сопротивления тканей с

помощью устройств обратной связи. Были изучены ценности использования компьютеров и специализированного программного обеспечения для создания 3D-анатомических и биологических моделей, а также их потенциальной роли в подготовке медицинского персонала.

3D-модель состоит из набора поверхностей, каждая из которых представлена совокупностью точек в декартовой системе координат (X, Y, Z). Возможны следующие разновидности методов, с помощью которых можно смоделировать 3D-модель:

- полигональное моделирование – сетка точек, соединенных между собой и образующих полигоны;
- бикубическое параметрическое моделирование патча – использование изогнутых линий (например, неоднородные рациональные B-сплайны) для формирования криволинейных поверхностей;
- конструктивная геометрия твердого тела – создание путем операций объединения/вычитания простых фигур;
- моделирование поверхности подразделения – моделирование с локальным уточнением деталей позволяет плоским участкам иметь простую конструкцию и изогнутые поверхности.

Сетка 3D-модели формирует каркас объекта из совокупности вершин и предоставляет информацию о форме и пропорциях предмета, а поверхности получают такие атрибуты как цвет и блики, с помощью которых графическое изображение воздействует на мысли и чувства, что способствует приближению 3D-модели к реальному объекту. Модель с сегментарными секциями является оптимальной, потому что наилучшим образом облегчает преподавание анатомии печени [4]. При моделировании 3D-модели необходимо уделить особое внимание количеству используемых полигонов, благодаря чему 3D-модель можно отнести к низкополигональной или высокополигональной.

Методы обработки поверхностей созданных 3D-моделей, в свою очередь, можно подразделить на:

- наложение текстуры – цифровое изображение, накладываемое на 3D-поверхность;
- рельефное преобразование – имитация углублений и возвышений с помощью карты в оттенках серого;
- сглаживание цветового перехода – интерполяция значений освещенности соседних граней сетки модели, придающая гладкий вид модели;
- наложение бликов – управление значением отражения смоделированного объекта;
- наложение карты прозрачности – использование карт серого цвета для управления прозрачностью участков модели.

Каждый из таких методов позволяет улучшить в определенной степени реальность 3D-модели. Для уменьшения затрат возможно использование сканирования необходимого участка для автоматического создания прототипа модели. Сейчас 3D-анатомические модели, напечатанные с использованием технологий – компьютерной томографии (КТ), магнитно-резонансной томографии (МРТ) или ультразвуковой, используются для обучения и планирования сложных медицинских процедур [5].

Для автоматизации создания 3D-моделей в трехмерном моделировании используются:

- 3D-сканеры – создание полигональной сетки от твердого объекта методом триангуляции расстояния от него до различных точек предмета;
- фотограмметрия – создание полигональной сетки из серии фотоснимков и выделение общих точек;
- МРТ или КТ-сканирование – получение данных для построения 3D-модели из серии снимков.

Физические модели, используемые в качестве замены реальной анатомии, не гниют и являются ценными учебными пособиями, но виртуальные 3D-модели могут легко и дешево обмениваться в электронном виде, хотя и являются дорогими для производства [6]. Стык

исследований в IT и медицине играет благоприятную роль для решения актуальных проблем, связанных со здоровьем пациентов. Взаимодействие с 3D-моделями облегчает пространственное изучение анатомии в организме, а также снижает риски хирургических ошибок, которые связаны с затратами на здравоохранение.

Доступность 3D-принтеров привела к взаимодействию виртуальных 3D-моделей с 3D-печатью в медицине, что подтолкнуло развитие медицинских приложений для работы с диагностическими системами, лучшими из которых являются КТ и МРТ.

Литература

1. Кузьмин А.В. Трехмерное моделирование и визуализация в медицине // Вестник Пензенского государственного университета. – 2015. – № 4(12). – С. 122–127.
2. Gross M.H. Computer graphics in medicine: from visualization to surgery simulation // ACM SIGGRAPH Computer Graphics Homepage archive. – 1998. – V. 32. – № 1. – P. 53–56.
3. Бельченко В.А., Притыко А.Г., Климчук А.В., Филиппов В.В. Черепно-лицевая хирургия в формате 3D. – М.: ГЭОТАР-Медиа, 2010. – 224 с.
4. Kong X., Nie L., Zhang H., Wang Z., Ye Q., Tang L. et al. Do 3D-Printing Models Improve Anatomical Teaching About Hepatic Segments to Medical Students? A Randomized Controlled Study // World J Surg. Springer International Publishing. – 2016. – V. 40. – P. 1969–1976.
5. The Future of 3D-Printing in Medicine [Электронный ресурс]. – Режим доступа: <https://www.itnonline.com/article/future-3-d-printing-medicine-0/> (дата обращения: 20.02.2018).
6. Шилько С.В., Рябченко Т.В., Гавриленко С.Л., Шалобаев Е.В. Применение компьютерной механики при подготовке САД-моделей для 3D-печати градиентных материалов и изделий // Сб. докладов Международного научно-практического симпозиума «Перспективы развития аддитивных технологий в Республике Беларусь». – 2017. – С. 194–207.

**Ибрагимов Дмитрий Игоревич**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: didimoner@gmail.com

УДК 004**ОБЗОР МЕТОДОЛОГИИ ПРЕДСТАВЛЕНИЯ КОНТЕНТА РАЗЛИЧНОГО ТИПА
В БРАУЗЕРНОЙ СРЕДЕ****Ибрагимов Д.И.****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 способы представления мультимедиа-контента в интернет-браузере, приводится описание видов мультимедиа-контента, а также, в качестве примера, приведено подробное описание работы с PDF-документами в среде браузера.

Ключевые слова: мультимедиа, трансляция, потоковое вещание, онлайн, контент, браузер, формат файла, веб.

Современный браузер, помимо того, что является инструментом для отображения веб-страниц, выполняет роль мощного мультимедийного комбайна, способного отображать практически все виды мультимедиа-данных: текст, изображения, аудио, видео, анимации и даже трехмерные объекты. Все это стало реальностью благодаря развитию глобальной сети Интернет, вследствие чего стала возможной передача больших объемов данных за короткий промежуток времени. Пользователь получает возможность просматривать колоссальное число цифровых галерей, использовать сервисы для прослушивания музыки, пользоваться потоковым вещанием видео, что позволяет смотреть любой видеоконтент без предварительной загрузки всего файла целиком, работать с документами, а также играть в полноценные трехмерные игры.

Помимо мультимедиа-контента, в браузер перебрались и классические приложения, такие как Word, Excel, Power Point, Photoshop и тому подобные. Данного шага невозможно было бы достигнуть только за счет увеличения скорости доступа в Интернет, поэтому здесь большую роль играет эволюция языка программирования JavaScript. Превращения js из простого инструмента для создания «динамики» на веб-странице в полноценный язык программирования позволило поместить в браузерную среду полномасштабные приложения, создание которых раньше было возможно лишь под конкретную операционную систему.

Огромным преимуществом веб-приложений является кроссплатформенность: для того, чтобы запустить приложение на любой операционной системе, достаточно иметь лишь современный веб-обозреватель и доступ к сети Интернет. Больше нет необходимости искать подходящий для вашей операционной системы дистрибутив, чтобы, скажем, просмотреть и отредактировать содержимое .doc документа, все это можно выполнить прямо в браузере, который имеется и на мобильных устройствах, что делает возможным запуск веб-приложений и на смартфонах [1].

Использование полноценной трехмерной графики в браузерной среде стало возможным благодаря появлению спецификации WebGL, являющейся «прослойкой» между JavaScript и

API OpenGL. Так же, как и OpenGL, WebGL является API низкого уровня, из чего следует, что для создания сложных проектов потребуются хорошо разбираться во многих аспектах трехмерной графики. Получение доступа к такой мощной графической системе напрямую из браузера открыло разработчикам новые возможности для создания комплексных систем, а также портирования уже имеющихся приложений и игр. Например, очень популярный в свое время Quake III (1999 г., id software) был успешно перенесен в браузер в 2010 году, выступив примером удавшегося переноса крупной игры [2].

Стоит сказать о том, что корпорация Google выпустила и активно поддерживает операционную систему Chrome OS, полностью основанную на одноименном браузере. В ней возможна установка приложений лишь из магазина Chrome (все приложения в нем написаны для браузерной среды), при этом устройства на базе данной операционной системы (ОС) успешно заменяют ноутбук с «взрослыми» ОС для выполнения повседневных задач (просмотр интернет-страниц, работа с документами, работа с мультимедиа). Такой подход компании наталкивает на мысль, что браузер в наше время стал чем-то гораздо большим, чем просто веб-обозревателем.

Наиболее универсальным форматом электронных документов является формат PDF (Portable Document Format), разработанный компанией Adobe. Основным способом создания PDF-документа является виртуальный принтер, предоставляющий возможность «распечатать» файл в электронный документ. Оригинальный документ готовится в специализированной программе (текстовый процессор, графический редактор, САПР), после конвертируется в формат PDF путем отправки на виртуальный принтер. Пользователь может быть уверен, что полученный электронный документ будет выглядеть одинаково на любой машине вне зависимости от версии программы-просмотрщика [3].

Веб-браузер полностью поддерживает работу с форматом PDF, имея для этого встроенные средства просмотра, но, если разработчику необходимо провести манипуляции с электронным документом на программном уровне, их будет недостаточно. На помощь приходят библиотеки для языка программирования JavaScript, занимающиеся обработкой документа и предоставляющей API (Application Programming Interface) для взаимодействия с ним. Можно привести несколько примеров таких библиотек: Viewer.js, PDFKit, PDF.js, но лучшей среди них является PDF.js, разработанная компанией Mozilla.

PDF.js – это JavaScript-библиотека, предназначенная для рендеринга PDF-документов средствами HTML5 Canvas. Библиотека может быть использована как часть веб-сайта, либо как расширение для браузера, позволяя открывать PDF-документы, расположенные на машине пользователя. PDF.js можно поделить на три уровня:

1. ядро. На этом уровне происходят операции парсинга и обработки документа. Уровень ядра является основой для всех остальных уровней. Не рекомендуется использовать методы ядра напрямую, так как для этого имеется API с высшим уровнем абстракции;
2. отображение. Данный уровень, используя уровень ядра, предоставляет удобный API для работы с документами и отображением информации из них;
3. просмотрщик. Заключительный компонент библиотеки, задачей которого является предоставление пользовательского интерфейса (UI – User Interface), при помощи которого пользователь будет манипулировать документами.

Встроенный просмотрщик (рисунок) является лишь примером того, как может быть построен интерфейс для работы с библиотекой, поэтому создатели рекомендуют создать свое решение на основе имеющегося, чтобы максимально соответствовать внешнему виду веб-сайта [4].

Для того чтобы начать использовать библиотеку в собственном проекте, необходимо лишь скачать ее исходный код и добавить в проект, либо воспользоваться услугами бесплатных CDN (Content Delivery Network), позволяющих подключать библиотеки без предварительной загрузки исходного кода в проект. После выполнения данного шага

достаточно инициализировать библиотеку и указать путь к документу, который следует открыть: `PDFJS.getDocument('helloworld.pdf')`.

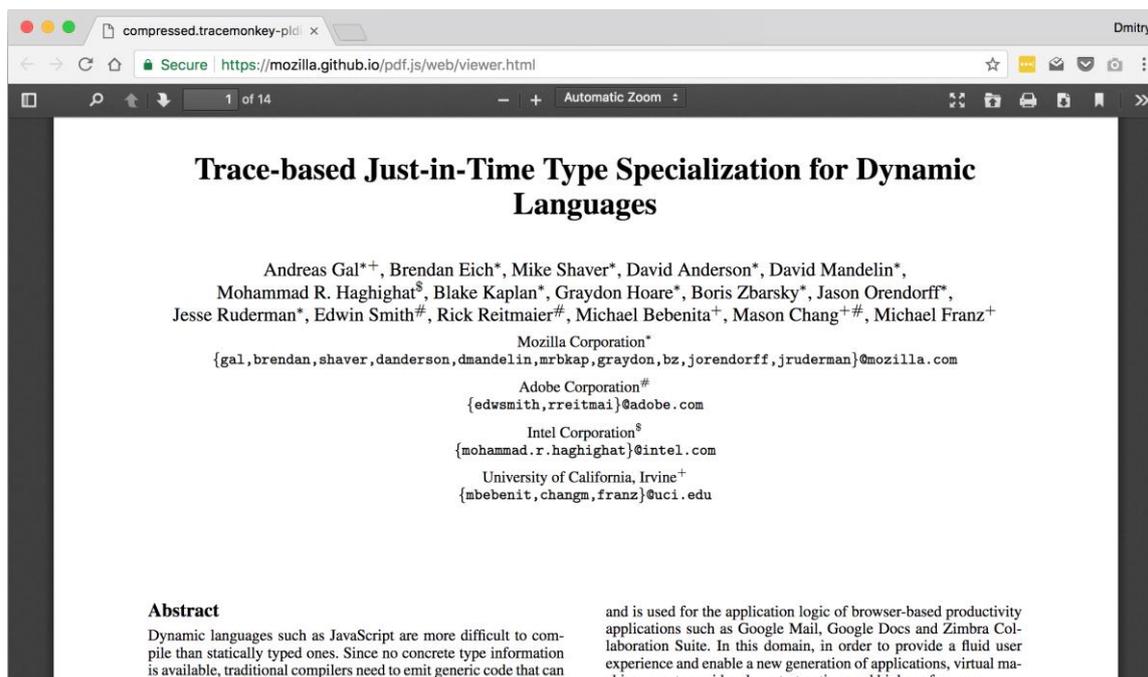


Рисунок. Внешний вид стандартного просмотрщика PDF.js

В заключение следует отметить, что тенденция переноса приложений и сервисов в веб с каждым годом набирает обороты, поэтому разработка приложений для этой среды является достаточно перспективным решением.

Литература

1. Семь принципов создания современных веб-приложений [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/242429/>, своб.
2. WebGL [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/WebGL>, своб.
3. Portable Document Format [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Portable_Document_Format, своб.
4. PDF.js [Электронный ресурс]. – Режим доступа: https://mozilla.github.io/pdf.js/getting_started/, своб.



Иванова Полина Владимировна

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: Polina-jdnk@yandex.ru

УДК 004.5

АНАЛИЗ ИССЛЕДОВАНИЙ ИНТЕРАКТИВНЫХ КНИГ ДЛЯ ДЕТЕЙ

Иванова П.В.

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 проанализированы исследования российских и зарубежных ученых, посвященных изучению литературы для детей дошкольного возраста и ее цифровому воплощению. Одни авторы исследовали в своих работах сам процесс чтения и методы передачи различного режима информации. В работах других авторов рассмотрена взаимосвязь между вербальными и визуальными элементами в интерактивной книге. Также рассмотрено положительное и негативное влияние интерактивных книг в процессе образования детей дошкольного возраста.

Ключевые слова: интерактивные технологии, воспитание и развитие, дизайн, книги для детей, дети дошкольного возраста, электронная книга, мобильное приложение.

Дошкольное детство, как период в человеческой жизни, играет важную роль в формировании того, каким станет человек в будущем. Заложенные в этот период образовательные, мировоззренческие, нравственные, культурные и физические приоритеты определяют жизненный путь. Чтение детских книг развивает художественно-речевые навыки, формирует нравственность и культуру ребенка, передает представления о жизни, труде, развивая социальный опыт и трудовую деятельность. В XXI веке происходит активное внедрение компьютерных технологий во все сферы деятельности. Эта тенденция коснулась также и книжного дела. Материал в таком виде доступен, удобен в использовании, экономичен и, следовательно, востребован.

Использование компьютерных технологий в образовательной деятельности позволяет перейти от объяснительно-иллюстрированного обучения к деятельностному, при котором ребенок становится активным субъектом, а не пассивным объектом педагогического воздействия, что способствует осознанному усвоению знаний дошкольниками.

В исследованиях G.R.Kress и L.Van [1, 2] понятие «чтение» выходит далеко за рамки простого распознавания текста. Под чтением понимается процесс восприятия знаков, которые могут быть представлены в различных режимах: текстовом, визуальном, слуховом, кинетическом и прочих. Чтобы развить у ребенка умение читать символы разных режимов в литературу следует включить, помимо классического текста, мультимодальный. На цифровом носителе возможны большее разнообразие и более комплексные комбинации. Слова и изображения могут быть объединены с движением, звуком и интерактивностью, под которой подразумевается вмешательство читателя в ход событий.

В своем исследовании Ольга Ро [3] представила цифровое воплощение детской литературы своеобразным «пятым элементом», средой, не противоположной, но объединяющей остальные четыре составляющие хорошей книги: сюжет, язык, иллюстрации и полиграфию.

L. David [4] описал динамику интерактивной книги как экосистему, где различные текстовые и визуальные составляющие порождают сложные отношения. Эти отношения порой имеют большее значение, чем несет в себе каждый элемент в отдельности. Если изменить один из элементов, то изменится восприятие и всего повествования в целом. Например, если историю проиллюстрирует другой художник, то результат получится новым и уникальным. В интерактивных приложениях эта экосистема еще больше расширяется, так как добавляются новые переменные в виде звука, анимации и интерактивности.

В исследовании L. Sipe [5] рассмотрены возможные преимущества интерактивных приложений перед печатным вариантом книги. Прежде всего, по мнению L. Sipe, важным преимуществом некоторых приложений является возможность чтения вслух. Таким образом, ребенок может одновременно следить за текстом и слушать запись диктора.

В некоторые приложения (L. Oamal [6]) также входит возможность звукозаписи. Функция записи позволяет родителям записывать чтение рассказа своим голосом, что позволит ребенку слушать книги в любой момент, даже если родителей не будет рядом, чтобы почитать лично. Данная функция также будет полезна на следующем этапе, когда ребенок начнет учиться читать самостоятельно, так как появится возможность записывать свой личный результат. Кроме того, данная функция поможет учителям и родителям проследить за прогрессом ребенка, сравнить навык его чтения в разные периоды времени, что служит ценным инструментом для оценивания прогресса.

В других приложениях данная функция выполняет несколько иную роль. Здесь детям предлагается рассказать (и записать голос в программе) классические сказки своими словами. Сами сказки представлены в приложении только визуально посредством иллюстраций, без текста.

В своем исследовании A.Y. Ghada [7] характеризует различные носители, на которых могут быть установлены интерактивные приложения. Для настольных компьютеров также выпускают интерактивные книги, но основной недостаток данных устройств выражен в том, что в таком случае требуются управления посредством мыши и клавиатуры, что ограничивает доступ к нему для маленьких детей. Телефон, из-за своего компактного размера не имеет возможности предоставить изображение в достаточном для детей дошкольного возраста масштабе. Размер же и вес планшетов делает цифровые приложения более похожими на печатную книгу, а сенсорный экран позволяет напрямую вводить команды, не используя мышь, что являются важными преимуществами.

До сих пор ведутся споры о том, стоит ли внедрять мультимедийные технологии в дошкольное образование.

В пользу интерактивных приложений следует выделить следующее:

- развитие у детей способности ориентироваться в информационных потоках окружающего мира;
- повышение интереса к обучению;
- обучение через игру;
- повышение навыка самоконтроля и самостоятельной деятельности;
- развитие интеллектуальных и творческих способностей;
- обеспечение личностно-ориентированного подхода;
- предоставление свободного выбора также влияет на формирование сознания ребенка в целом;
- обеспечение личностной значимости усваиваемых знаний;
- расширение возможностей родителей и педагогов в сфере раннего развития;
- автоматизированный контроль правильности освобождает время педагога для параллельной работы с другими детьми;
- формирование двигательных навыков и концентрации, также развитие координации движений рук.

При этом необходимо выделить и некоторые отрицательные факторы:

- неблагоприятное влияние планшетных устройств на физическое и психическое состояние ребенка;
- ориентирование на эмоциональное, а не логическое осмысление материала;
- обилие мультимедийных материалов может отвлекать внимание детей от содержания книги;
- приучение к поверхностному, развлекательному чтению;
- ограниченное санитарными нормами время взаимодействия с гаджетами для ребенка.

В заключение следует отметить несколько важных моментов, о которых не следует забывать, если принято решение включить электронные устройства в работу с детьми:

- они не могут заменить эмоционального человеческого общения, так необходимого в дошкольном возрасте;
- это лишь средства, дополняющие педагога, а не заменяющие его;
- все средства должны удовлетворять санитарным нормам;
- должна быть граница между физической и виртуальной реальностью;
- это не средство досуга, а средство познания и исследования окружающего мира.

В процессе выполнения работы были проанализированы зарубежные и отечественные источники по теме исследования, выявлены наиболее и наименее проработанные аспекты в области изучения интерактивных книг для детей дошкольного возраста.

В результате анализа работ в данной области тема будущего исследования была сужена с «интерактивных книг» до «интерактивных книг-приложений», так как была выявлена несообразность использования персонального компьютера для обучения детей дошкольного возраста в связи с неудобностью и сложностью использования. Интуитивный дизайн и управление, размер и эргономика планшета подходит для детей гораздо больше, одновременно не теряя в функционале, нужном для данной продукции.

Литература

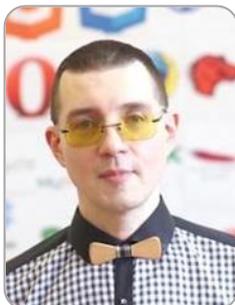
1. Kress G.R. *Multimodality: A social semiotic approach to contemporary communication*. – London: Routledge, 2010. – 212 p.
2. Van Leeuwen T. *Introducing social semiotics*. – London: Taylor & Francis e-Library, 2005. – 314 p.
3. По О. *Детская книга и Интернет: реальность цифровая, альтернативная и дополненная [Электронный ресурс]*. – Режим доступа: <http://pro-books.ru/sitearticles/3513>, своб.
4. Lewis D. *Reading contemporary picturebooks: Picturing text*. – New York; London: Routledge Falmer, 2001. – 200 p.
5. Lawrence S. *How picture books work: A semiotically framed theory of text-picture relationships // Children's Literature in Education*. – 1998. – V. 29(2). – P. 97–108.
6. Oamal L.U. *Tiny Great True Stories (book-app)*. – Version 1.1. – EditorCaixote, 2015. Available for iOS.
7. Al-Yaqout G. *From Slate to Slate: What does the Future Hold for the Picturebook Series? // New Review of Children's Literature and Librarianship, Abingdon*. – 2011. – V. 17(1). – P. 57–77.

**Казначеева Екатерина Олеговна**

Год рождения: 1993

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207Направление подготовки: 09.04.02 – Информационные системы
и технологии

email: kattya.kazn@gmail.com

**Государев Илья Борисович**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: goss@corp.ifmo.ru

УДК 004**МОДУЛЬНЫЙ И КОМПОНЕНТНЫЙ ПОДХОДЫ В ПРОГРАММИРОВАНИИ****Казначеева Е.О., 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 два подхода, используемые при разработке современных веб-приложений: модульный и компонентный. Целью обоих подходов является компоновка написанного кода для повторного использования. Рассмотрены каждый из упомянутых подходов, а также выделены их общие черты и различия. Сделан вывод о дальнейшем использовании модулей и компонентов при разработке веб-приложений.

Ключевые слова: JavaScript, Node.js, модуль, компонент, веб-приложение.

В настоящее время сложно представить JavaScript-код, не разбитый на отдельные элементы, которые как «по полочкам» разложены в проекте, и впоследствии подключаются и используются для создания нового функционала, необходимого для разработки. Подобная структура первым делом позволяет минимизировать дублирование написанного кода. Методология повторного использования кода в JavaScript разделилась на модульный и компонентный подходы, которые имеют как общие, так и расхожие черты, что породило проблему многообразия вариантов подключения отдельных элементов к проекту.

Модульная архитектура стала неотъемлемой частью разработки веб-приложений. Изначально скрипты были небольшие и не требовали разделения на модули, но со временем JavaScript стал привлекать внимание все большего числа разработчиков веб-приложений, как frontend, так и backend программистов (Node.js), и написанные скрипты становились все объемнее, их с трудом можно было поддерживать, что привело к необходимости разделения кода на отдельные составляющие – модули.

Модуль – это функционально независимый элемент кода, который выносится в отдельный файл. Впервые модуль, как самостоятельная единица кода, был предложен Ричардом Корнфордом в 2003 году и был основан на немедленно вызываемой функции, которая образует приватную область видимости:

```
var MODULE = (function() {
```

```
var privateVariable = 1;
function privateMethod() {
  // ...
}
return {
  moduleProperty: 1,
  moduleMethod: function()
  // ...
}
};
}());
```

Называлась такая структура паттерн «Модуль». Далее при попытке создания серверного JavaScript (CommonJS) была предложена иная реализация вышеописанного паттерна, которая состояла из двух частей: экспорт сущности с помощью структуры `module.exports` и ее импорт с помощью функции `require()` [1]. В то же время велись обсуждения по асинхронной загрузке модулей в браузере, по результатам которых был предложен иной формат работы с модулями – AMD (Asynchronous Module Definition). В основе данного формата лежат функции `define()` и `require()` для экспорта и импорта модулей соответственно. Из-за того, что одновременно использовались два формата для одних и тех же целей, в 2011 году появился формат UMD (Universal Module Definition), объединяющий два вышеперечисленных формата. Официальная поддержка модулей в JavaScript появилась в стандарте ES6 в 2015 году [2]. Нативные модули могут экспортировать и импортировать свои части, используя специально зарезервированные для этого слова `export` и `import`.

Ключевое слово `export` можно ставить:

- перед объявлением переменных, функций и классов;
- отдельно, при этом в фигурных скобках указывается, что именно экспортируется.

При помощи ключевого слова `as` можно указать, что переменная будет доступна снаружи (экспортирована) под указанным именем. Другие модули могут подключать экспортированные значения при помощи ключевого слова `import`.

Как правило, код стараются организовать так, чтобы каждый модуль отвечал за одно действие. Иначе говоря, «один файл – одна сущность, которую он описывает». Для такой ситуации, когда один модуль экспортирует одно значение, предусмотрено особое ключевое сочетание `export default`. Если поставить после `export` слово `default`, то значение станет «экспортом по умолчанию». Такое значение можно импортировать без фигурных скобок.

Модули позволяют ограничить область видимости, позволяют повторно использовать части приложения, делают приложение более структурированным и вообще делают код нагляднее. Более того, модули можно тестировать отдельно от всего приложения, что значительно сокращает время тестирования проекта.

При разработке современного веб-приложения одними из наиболее важных аспектов являются модульность и расширяемость [3]. При создании грамотной модульной архитектуры добавление нового элемента в проект не составит большого труда.

Модульная архитектура приложения сравнима с файловой системой компьютера. Разделение файлов по директориям с особыми наименованиями формирует структуру, в которой легко разобраться человеку, который сталкивается с данной системой впервые.

Другим способом структурирования кода являются компоненты. Организация приложения в виде набора компонентов – это мощное средство программирования, позволяющее программисту справляться со всеми более сложными задачами. Путем разделения программы на независимые компоненты программист может понизить видимый уровень ее сложности. При компонентно-ориентированном подходе программа организуется как набор компонентов, которые можно использовать, не вникая в детали их внутренней

реализации. Суммарный эффект такого подхода состоит в снижении общей сложности программы.

Компоненты позволяют собирать большие приложения из меньших кусочков. Они представляют собой пригодные к повторному использованию объекты. Практически любой интерфейс можно представить как дерево компонентов.

Компоненты расширяют базовые HTML-элементы, позволяя инкапсулировать повторно используемый код. Не вдаваясь в подробности, можно сказать, что компоненты – это пользовательские элементы, к которым фреймворк привязывает определенное поведение.

Для работы с компонентами используются различные библиотеки и фреймворки, например, React и Vue.js.

После появления нативных модулей в JavaScript, также появилась и возможность подключать модули к веб-странице, не используя сторонних систем сборки, а пользуясь только возможностями HTML и JavaScript (`<script type=«module»>`). Однако на сегодняшний день данная технология поддерживается не во всех браузерах.

Использование JavaScript-модулей в настоящее время становится все более распространенным, все чаще в проектах можно встретить использование нативных модулей. Многочисленные статьи и посты по теме нативных модулей могут свидетельствовать об актуальности исследований в данной области [4]. То же самое можно сказать и про компоненты. Популярность компонентно-ориентированных библиотек и фреймворков говорит о том, что и дальше компоненты будут использоваться при разработке современных веб-приложений.

Подводя итог, стоит сказать, что нативные модули JavaScript и веб-компоненты появились сравнительно недавно, но уже уверенно занимают свои позиции и вытесняют технологии, которые разработчики использовали из необходимости. Вскоре при разработке веб-приложений можно будет отказаться от сторонних технологий по сборке модулей в единый файл и задействовать только нативные средства данного языка программирования.

Литература

1. Модули в JavaScript [Электронный ресурс]. – Режим доступа: <https://getinstance.info/articles/javascript/javascript-modules/> (дата обращения: 31.01.2018).
2. Axel Rauschmayer Exploring ES6. Upgrade to the next version of JavaScript. – Leanpub, 2015. – 555 p.
3. Modular Application Architecture – Intro [Электронный ресурс]. – Режим доступа: <https://www.goetas.com/blog/modular-application-architecture-intro/> (дата обращения: 31.01.2018).
4. Нативные ECMAScript модули – первый обзор [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/company/tuturu/blog/326716/> (дата обращения: 31.01.2018).



Калинина Кристина Владимировна

Год рождения: 1993

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4205

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: tiiinaaaliii@gmail.com



Шуклин Дмитрий Анатольевич

Год рождения: 1979

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: do@limtu.ru

УДК 004.92

**ПОДХОДЫ К ПРОЕКТИРОВАНИЮ ВЕБ-СЕРВИСА ПО АРЕНДЕ
НЕДВИЖИМОСТИ**

Калинина К.В., Шуклин Д.А.

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

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

В работе рассмотрены процессы сдачи и снятия квартиры в аренду. Рассмотрены стандартные схемы процессов сдачи и снятия квартиры в аренду. В ходе рассмотрения стандартных схем выявлены недостатки каждой из них. На основе рассмотренных схем предложено разработать сервис для аренды квартиры, который поможет решить выявленные недостатки и упростит процесс сдачи и снятия квартиры в аренду.

Ключевые слова: аренда квартиры, сервис для аренды квартиры, процесс сдачи квартиры в аренду, процесс снятия квартиры в аренду, стандартный процесс аренды квартиры, оптимизированный процесс аренды квартиры.

В настоящее время процесс аренды квартиры является актуальным для многих людей. И этот процесс можно рассматривать с двух точек зрения: со стороны владельца квартиры и со стороны того, кто хочет снять квартиру (или арендатора).

Если владелец квартиры хочет сдать ее, то на различных сервисах или в социальных сетях он может поместить объявление по аренде квартиры. Тогда для него процесс сдачи квартиры в аренду будет следующий:

1. владелец размещает объявление о сдаче квартиры на каком-либо сервисе;
2. ждет звонков от арендаторов и после встречается с одним из них для показа, оплаты и заключения договора. При этом заключение договора осуществляется не всегда, как и уплата налогов за аренду.

В такой схеме есть определенные риски для владельца квартиры, так как арендатор никаким образом не проверяется:

1. арендатор может попасться очень шумным или нечистоплотным, и будут поступать жалобы от соседей;
2. арендатор может оказаться мошенником. Снять квартиру на небольшой срок, а потом пересдать ее по поддельным документам другому человеку, взяв оплату на несколько месяцев вперед;

3. арендатор может снять квартиру для себя, но в итоге подселить еще несколько человек;
4. внезапный отъезд арендатора без предупреждения, когда еще не истек срок аренды квартиры;
5. несвоевременная оплата. Также арендатор может оплачивать аренду, но не платить за коммунальные услуги;
6. проблемы с налоговой службой. Кто-нибудь может настучать налоговой службе на то, что квартира сдается, и тогда для владельца это грозит штрафом [1].

С другой точки зрения для того, кто хочет снять квартиру, процесс будет следующий:

1. арендатор регистрируется на сервисе;
2. сервисы могут предоставлять платный доступ к базе данных квартир, тогда арендатору необходимо оплатить подписку;
3. по выбранным фильтрам и параметрам арендатор осуществляет поиск соответствующей квартиры;
4. если квартира подходит, то он сам звонит владельцу, узнает, свободна ли квартира, договаривается о встрече и просмотре квартиры;
5. если арендатора все устраивает, то он оплачивает аренду, подписывает договор (не всегда) и начинает проживание в квартире [2].

Такая схема имеет большие риски для арендатора, так как на сервисах, даже с платной подпиской, недостаточно тщательно проверяют объявления о сдаче квартир:

1. арендатор может связаться не с владельцем квартиры, а с агентом, который возьмет за свои услуги до 100% комиссии;
2. владелец квартиры может оказаться подставным лицом, который тоже снял квартиру на какой-то срок, и пересдает ее по завышенной цене;
3. в объявлении указаны другие фотографии. Пустая трата времени на дорогу до квартиры;
4. конторы, которые предлагают заплатить за их услуги в офисе и дают адрес квартиры и телефон владельца. Приезжаете на место, а там никто не открывает [3];
5. в большинстве случаев сдача квартиры происходит незаконно, без уплаты налогов государству;
6. в таких случаях не всегда заключается договор, и никто не может гарантировать, что на следующий день владелец квартиры не приведет новых жильцов;
7. у квартиры может быть несколько владельцев. Другие собственники могут явиться в любой момент и выселить [4].

Рассмотренные процессы аренды квартиры имеют риски для владельца квартиры и для того, кто хочет снять квартиру. Чтобы исключить все выше перечисленные недостатки, предложено разработать сервис аренды квартир, который должен решать следующие задачи:

- позволять снять квартиру;
- позволять сдать квартиру;
- иметь возможность для владельца выбора дополнительных услуг по сдаче квартиры;
- иметь возможность для владельца отслеживать онлайн-состояние квартиры и выплаты по квартире;
- позволять выплачивать налоги государству;
- иметь возможность проверки арендатора на платежеспособность и законопослушность;
- гарантировать проверку квартиры.

С помощью данного сервиса для владельца квартиры схема работы будет следующей:

1. владелец квартиры подает заявку на сдачу квартиры, и с ним связывается менеджер для уточнения всех деталей;
2. при подаче заявки он может выбрать дополнительные услуги (уборка квартиры, страхование и другие);
3. владелец квартиры приезжает в офис и заключает договор с компанией. После этого он становится их клиентом, и далее от лица владельца действует компания.

Преимущества работы владельца с сервисом следующие:

1. все арендаторы проверяются компанией;
2. выбор дополнительных услуг без привлечения сторонних компаний для каждой из требуемых услуг;
3. в этой схеме всегда производится выплата налогов государству, что гарантирует законность процесса сдачи квартиры;
4. владельцу квартиры достаточно лишь один раз приехать в офис для заключения договора, далее он может находиться в любом месте и отслеживать все выплаты и состояние квартиры через личный кабинет.

С помощью данного сервиса человек может снять квартиру по следующей схеме:

1. арендатор подает заявку на доступ к личному кабинету, где содержится база квартир. После этого с ним связывается менеджер для уточнения всех деталей;
2. оплачивает подписку и получает доступ к базе квартир;
3. арендатор начинает поиск квартиры по соответствующим фильтрам и параметрам;
4. после выбора квартиры арендатор бронирует ее, после чего может посмотреть квартиру. И если его все устраивает, то заключает договор на аренду, производит оплату и начинает проживание в квартире.

В такой схеме есть ряд преимуществ для арендатора:

1. все владельцы квартир и необходимые документы на квартиру детально проверяются компанией;
2. в базе данных сервиса содержатся только актуальные предложения и их полная стоимость;
3. заключается договор аренды, который дает гарантии арендатору, что он будет жить в выбранной квартире на период, указанный в договоре;
4. вся необходимая информация содержится в личном кабинете арендатора;
5. в такой схеме отсутствуют постоянные визиты владельца квартиры. Все просмотры осуществляются по предварительной согласованности.

В результате анализа стандартного процесса по аренде квартиры были выявлены его недостатки. Разрабатываемый сервис позволит решить эти недостатки и упростить процесс.

Литература

1. Как снизить риски, сдавая жилье в аренду? [Электронный ресурс]. – Режим доступа: <http://www.gazeta.bn.ru/articles/2011/09/12/85911.html>, своб.
2. Клименко Е.А. Аренда квартиры: методы поиска и связанные с этим риски // Международный научный журнал «Инновационная наука». – 2015. – № 12. – С. 130–131.
3. Опасная аренда. Как снять квартиру без риска быть обманутым [Электронный ресурс]. – Режим доступа: <http://www.aif.ru/realty/price/1115179>, своб.
4. Феоктисов А.В., Карамышева Ю.В. Нелегальная аренда жилых помещений: проблема развития и перспективы устранения // Электронный научный журнал «Наука. Общество. Государство». – 2015. – № 1(9). – С. 1–10.

**Кардашов Сергей Алексеевич**

Год рождения: 1996

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4107Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: krosh961@yandex.ru

**Сокуренко Юрий Андреевич**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», к.т.н., доцент

e-mail: kpd@limtu.ru

**Государев Илья Борисович**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: goss@corp.ifmo.ru

УДК 004.92**СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОЛОГИЙ ПРОТОТИПИРОВАНИЯ
ВЕБ-РЕСУРСОВ****Кардашов С.А., Сокуренко Ю.А., Государев И.Б.****Научный руководитель – к.т.н., доцент Сокуренко Ю.А.**

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

В работе проведен анализ методологий для прототипирования веб-ресурсов. Рассмотрены исследования по изучению фундаментальных принципов выбора методологий прототипирования IT-продуктов, методов анимирования веб-интерфейсов в прототипах и подходов к выбору технологий для реализации анимации.

Ключевые слова: разработка веб-сайтов, инструменты разработки сайтов, веб-интерфейс, веб-приложение, прототипирование.

Современные компании по разработке и проектированию веб-приложений интерфейсов пользователя каждый день сталкиваются с вопросом как визуализировать и анимировать прототипы для более точного понимания как спроектированный интерфейс должен быть анимирован в различных ситуациях.

На сегодняшний день в разработке web-проектов одним из главных этапов является прототипирование. Прототипирование как процесс создания программного обеспечения (ПО) условно делят на две части: прототипирование пользовательского интерфейса и проектирование функциональных требований [1]. Для проектирования функциональных требований используются средства моделирования UML и IDEF0, которые являются

промышленными стандартами при разработке ПО [2]. В прототипировании пользовательского интерфейса веб-ресурсов стандартов нет. Есть отдельные советы, методы и условия.

Актуальность темы обосновывается тем, что по мере развития и внедрения новых технологий, переосмысливаются и методы к прототипированию интерфейсов. Одновременно происходят существенные изменения в использовании веб-сайтов. Изменения технологий инициируют изменения методов разработки инструментов для прототипирования. Новые технологии позволяют информационным системам генерировать более эффективные сценарии взаимодействия. С одной стороны, технологии открывают новые возможности для проектирования веб-интерфейсов. С другой стороны, требования к современным веб-интерфейсам постоянно растут и тем самым заставляют разработчиков совершенствовать методы прототипирования и подстраиваться под стоящие перед ними задачи.

Правильное размещение элементов интерфейса повышает его юзабилити, делает ресурс более привлекательным для пользователей [3]. В отдельных случаях, с помощью грамотной компоновки сайта можно склонить посетителей к определенным действиям: оставить заявку, подписаться на рассылку, приобрести товар и т.п.

С этой целью на этапе проектирования предварительно создают прототип – черно-белый макет, представляющий упрощенную схему сайта. Он содержит все основные элементы, представленные в виде блоков, поэтому клиент может оценить основную концепцию.

Прототипирование помогает не только выявить и сформулировать основное направление дизайна, но и значительно сэкономить время. Потратив день на создание концепта, вам не придется тратить неделю на разработку нового макета сайта. Но это не единственная причина, по которой стоит уделить время прототипу.

Прототип – это схематичный набросок страницы или страниц сайта с изображенными на нем элементами дизайна (кнопками, изображениями, меню, формами).

В процессе разработки прототипирования интернет-сайтов были изучены работы по проектированию и моделированию информационных систем и по разработке интернет-сайтов.

На момент исследования было выявлено, что статичное прототипирование без интерактивной базы очень плохо воспринимается людьми, которые мало связаны со сферой IT-разработок. Для лучшего результата визуализации прототипов применяются технологии анимирования статичных элементов, что, в свою очередь, дает больший отклик на предоставленный прототип веб-ресурса.

Инструменты, применяемые при анимировании элементов прототипа, часто уже встроены в ПО, которые предоставляет возможность создавать статичные прототипы.

ПО для создания прототипов может быть как веб-приложением, так и приложением, устанавливаемым на персональный компьютер.

При анализе были взяты два ПО Principle и Invision. Оба продукта имеют свои недостатки, которые связаны с ограничением использования на определенных операционных системах, возможностью работать с исходными данными разных продуктов для создания прототипов и способностью работать автономно.

Основной проблемой является то, что те или иные ПО не позволяют работать с исходными данными любых других продуктов для прототипирования, а предлагают использовать только встроенные средства для прототипирования и анимирования.

Таким образом, следует отметить, что на момент проведения анализа методологий прототипирования веб-ресурсов существуют решения для анимирования прототипов, но они имеют существенные недостатки, которые ограничивают пользователей использовать различные программные средства для прототипирования и предлагают только свои решения.

Литература

1. Прикладные информационные системы: сборник научных трудов / Под ред. Е.Н. Эгова. – Ульяновск: УлГТУ, 2014. – 271 с.
2. Слива М.В. Прототипирование графического интерфейса пользователя как неотъемлемая часть процесса разработки программного обеспечения // Вестник НВГУ. – 2013. – № 1. – С. 74–76.
3. Прототипирование и его роль в разработке сайта [Электронный ресурс]. – Режим доступа: https://depix.ru/articles/prototipirovanie_i_ego_rol_v_razrabotke_sayta (дата обращения: 11.12.2017).



Катафеев Дмитрий Юрьевич

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4208

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: dima.katafeev@gmail.com



Погорелов Виктор Иванович

Год рождения: 1940

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, д.т.н., профессор

e-mail: pogvic@mail.ru

УДК 004.9

**ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ МЕТОДОВ ГЕЙМИФИКАЦИИ В СИСТЕМАХ
ЗДРАВООХРАНЕНИЯ**

Катафеев Д.Ю.

Научный руководитель – к.т.н., доцент Погорелов В.И.

Работа выполнена в рамках темы НИР № 643202 «Исследование методов и алгоритмов формирования облачных информационных систем».

В работе рассмотрены основные способы использования средств и методов структурной геймификации, а также способы внедрения их в системы здравоохранения, с целью повышения мотивации у граждан вести здоровый образ жизни.

Ключевые слова: геймификация, технологии геймификации, игровые технологии, геймификация в медицине, игра, здоровье.

В последние годы все мы наблюдаем широкое развитие информационных технологий абсолютно в различных сферах деятельности: в обучении, в здравоохранении, в производстве. Это влечет за собой использование средств геймификации, например, в медицине. Ведь только за последний год около семидесяти процентов пользователей искали в сети информацию, связанную со здоровьем. Основные проблемы оказания медицинских услуг являются длинные очереди, нехватка времени на работу с пациентом и др., поставили перед Министерством здравоохранения задачу: совершенствовать между врачом и пациентом процессы обмена информацией с помощью внедрения информационных технологий [1].

Современная молодежь забыла, что такое здоровый образ жизни. Дети и подростки колоссальное количество времени проводят за компьютерами или другими гаджетами, ведут малоподвижный образ жизни, тем самым провоцируя развитие многих болезней, связанных с опорно-двигательным аппаратом. Для решения подобной проблемы поможет применение структурной геймификации.

Пожалуй, стоит объяснить, что же такое структурная геймификация, о которой идет речь в данной работе. Существуют два вида геймификации. Это структурная и содержательная. При содержательной геймификации игровые элементы и игровое мышление применяются для изменения содержания, чтобы предоставить его в виде игры. В

структурной геймификации игровые элементы применяются без каких-либо изменений [2]. Иными словами, структурная геймификация привносит игровые элементы, без изменения содержания и не является игрой, т.е. структурная геймификация подразумевает встраивание игровых элементов в систему без изменения ее содержания.

В игре содержится множество элементов мотивации, таких как состязательность, призовые стимулы, логика преодоления препятствий, свобода влиять на игровой мир, немедленный его отклик на любое действие игрока. Иначе говоря, любая деятельность не становится игрой, но заставляет думать, что происходит участие в игре. Внедрение игровых технологий позволит повысить мотивацию к ведению подвижного и здорового образа жизни [3].

Геймификация – это эффективный способ для увеличения вовлеченности пользователей, но важно применять комплексно, а не просто игровые атрибуты (бейджи, очки, награды и т.д.), и для каждой сферы разрабатывать индивидуальную систему. К примеру, в системе здравоохранения сейчас используются различные приложения («сканер еды», показывающее количество полезных и вредных веществ на тарелке) и гаджеты (фитнес-браслет, измеряющий пульс, пройденное расстояние и скорость, «умная зубная щетка»).

Используя подобные приложения, которые позволяют следить за количеством съедаемых полезных веществ, испытываемой нагрузкой на организм, количеством сожженных калорий, ритмом сердцебиения, пройденным расстоянием, пользователь может значительно снизить риск заболеваний, связанных с кишечно-желудочным трактом, поднять свой иммунитет и тем самым поправить свое самочувствие.

Актуальность данного исследования обусловлена быстрым развитием информационных технологий и их использованием в повседневной жизни, как видно медицинская деятельность не стала исключением. Игры активизируют выработку «гормона удовольствия», предлагая радость от одобрения и побед. Таким образом, создавая геймифицированную систему, можно превратить процесс выполнения задачи в удовольствие. Главное не сосредоточиться только на удовольствии от процесса, и не опираться на примитивные механизмы, так как все это элементы внешней мотивации. Можно создать внутреннюю мотивацию, т.е. сделать так, чтобы выполнение задач стало значимым само по себе.

Благодаря этому, с помощью геймификации возможно мотивировать людей регулярно заниматься спортом, есть здоровую пищу, рационально использовать природные ресурсы. Главной задачей для реализации таких проектов является внедрение таких технологий в медицинскую деятельность, и как с помощью нее повысить мотивацию. Именно эти задачи необходимо решить на начальных этапах.

Нельзя возлагать все надежды на геймификацию, ведь не во все рабочие процессы можно добавить элемент игры. В основном геймификация подходит для таких отраслей, сотрудникам которых близко само понятие игры. Например, современное поколение, воспитанное на компьютерных играх, охотно принимают такие нововведения. В свою очередь, не все пользователи старшего поколения могут понять смысл тех или иных действий, ведь, по их мнению, свои обязанности и так нужно выполнять во время рабочего процесса. Также нужно принять, что некоторые рабочие задачи не могут носить развлекательный характер. Следующим аспектом является анализ проблемы, в которую нужно внедрить элемент игры.

Геймификация должна использоваться там, где есть место нематериальным сторонам рабочей деятельности. Одной из главных составляющих геймификации является стремление человека быть оцененным.

В большинство рутинных процессов все-таки можно внедрить геймификацию, рассмотрим, на что нужно обратить внимание. Профессор Би Джей Фогг (B.J. Fogg), руководитель Persuasive Tech Lab Стэнфордского университета, предлагает модель

поведения (рисунок), состоящую из трех составляющих: мотивация, возможность, импульс. Чтобы повысить интерес и вовлеченность человека, все три элемента должны существовать в системе и действовать одновременно [4].

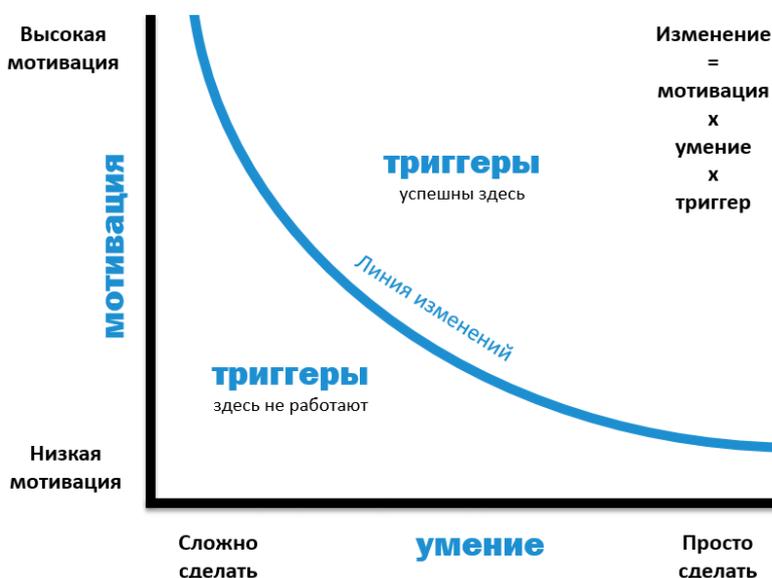


Рисунок. Модель поведения Фогга

Для того чтобы поддерживать мотивацию на длительном промежутке времени, необходимо завлечь пользователя, например, посредством введения в приложение систем подсчета очков, предмет поощрения, чтобы у пользователей появился интерес следовать предложенным инструкциям. Интеграция развлечений и игр с медицинским приложением должна мотивировать пациента и собирать информацию, необходимую для выработки более информированных решений для лечения и планирования его активностей, вместе с этим поможет собирать необходимые данные, для принятия обоснованных решений касательно ежедневной активности, которые способствуют улучшению состояния здоровья пациента.

Таким образом, после внедрения технологий геймификации в медицину, можно создать систему контроля здоровья в игровой форме, что будет удерживать пациента от пренебрежения соответствующим планом лечения. Совмещение развлечения и игры в приложении могут мотивировать пользователя, а также собирать необходимую информацию касательно ежедневной активности, рациона питания, здорового сна, для улучшения здоровья, четко следовать указаниям врача и проходить полный курс лечения.

Литература

1. Пациент – врач. Особенности коммуникации. Информатизация здравоохранения [Электронный ресурс]. – Режим доступа: http://www.remedium-journal.ru/arhiv/detail.php?ID=52840&num=%B910&sec_id=5489, своб.
2. Кэпп К.М., Blair L., Blair R. The Gamification of Learning and Instruction Fieldbook: Ideas into Practice. – John Wiley & Sons, 2013. – 480 p.
3. Зимина Д.В., Козак О.О., Погорелов В.И., Шуклин Д.А. Особенности реализации структурной геймификации на платформе WordPress для создания дистанционных обучающих курсов // Интернет-журнал «Науковедение». – 2016. – Т. 8. – № 3(34). – С. 117.
4. Fogg B.J. BJ Fogg's behavior model [Электронный ресурс]. – Режим доступа: <http://www.behaviormodel.org> (дата обращения: 25.02.2017).

**Кемалжанов Дамир Ильясович**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления «Академия ЛИМТУ», кафедра компьютерного проектирования и дизайна, студент группы № S4105

Направление подготовки: 09.04.02 – Информационные системы и технологии

e-mail: damir.kemalzhanov@mail.ru

**Романова Асель**

Год рождения: 1993

Университет ИТМО, факультет методов и техники управления «Академия ЛИМТУ», кафедра компьютерного проектирования и дизайна, аспирант

Направление подготовки: 09.06.01 – Информатика и вычислительная техника

e-mail: asel-romanova@mail.ru

УДК 004.451.83

**ТЕХНОЛОГИИ УДАЛЕННОГО ДОСТУПА К КОРПОРАТИВНЫМ
ИНФОРМАЦИОННЫМ СИСТЕМАМ****Кемалжанов Д.И., Романова А.****Научный руководитель – к.педагог.н., доцент Шуклин Д.А.****Консультант – аспирант Романова А.**

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

В работе рассмотрены технологии удаленного доступа к корпоративным информационным системам, современное состояние рынка информационных технологий, в частности к корпоративным мобильным приложениям. Выявлены ключевые требования для дальнейшего исследования технологии удаленного доступа к корпоративным информационным системам.

Ключевые слова: удаленный доступ, корпоративные информационные системы, мобильные приложения.

Четвертая информационная революция приводит современные корпоративные системы в состояние постоянной эволюции. С появлением мобильных сетей и мобильных технологий, возможность реализации удаленного доступа становится менее трудоемкой. Интеграция мобильных приложений к корпоративным информационным системам привела к созданию нового сектора разработки информационных решений на предприятиях разного уровня.

Рынок корпоративных мобильных приложений расширяется с каждым годом, с открытием программного кода мобильных платформ API (Application Programming Interface) расширились возможности кастомизации приложений под определенные задачи. В свою очередь, открытый программный код запускает целые механизмы воздействия на управление предприятием на самых разных уровнях повышением эффективности за счет средств информационных технологий. Одним из наиболее релевантных решений будет создание мобильного пространства и приложения для решения определенного ряда задач: предоставление базы данных, корпоративных знаний и др. Однако современный рынок предлагает «коробочные» варианты построения систем, которые несут за собой ряд сложностей, такие как отсутствие гибкости и масштабируемости [1].

Большинство корпоративных мобильных приложений построены на клиент-серверной архитектуре, обусловлено это тем, что основная часть функций приложения выполняется на

сервере в силу своей вычислительной мощности [2, 3]. Все это приводит такого рода систем к определенным требованиям, таким как:

- постоянный доступ к сети;
- высокая скорость Интернета, Интранета;
- структурированность данных;
- постоянная работоспособность сервера.

Современный рынок высоких информационных технологий приставляет широкий выбор смартфонов с многоядерными процессорами на базе архитектуры ARM (от англ. Advanced RISC Machine), большим объемом оперативной памяти и т.д. Что введет к широким возможностям использования смартфона вне зависимости от сервера [4].

Метод разделенных функции предлагает разделить систему на две части. Основная часть будет содержать серверное приложение, мобильное приложение на стороне клиента. Основным отличием от классической архитектуры является приложение-клиент, представленный на рисунке, в котором будет содержаться часть базы данных (БД(1)) и в зависимости от технических характеристик смартфона часть функций обработки данных будет переведена на процессор смартфона.

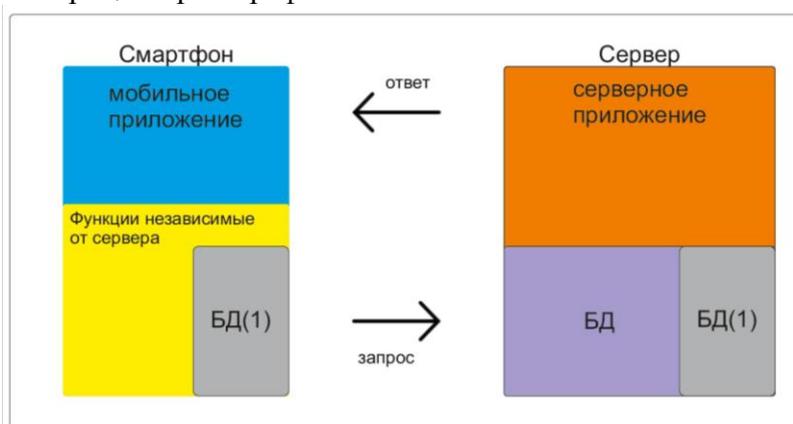


Рисунок. Архитектура клиент-серверного приложения

Система, построенная таким образом, обладает рядом положительных характеристик:

- частичная независимость от сети;
- возможность обработки информации удаленно;
- стабильность;
- масштабируемость;
- возможность оффлайн-режима.

Корпоративные мобильные приложения будут стремительно развиваться, в особенности в области большой промышленности. Построение мобильных приложений реализуется в частном порядке в зависимости от требований и функционала. Существует множество методов построения данной архитектуры, одним из них является «Метод разделенных функций». В дальнейших исследованиях будут проведены более подробные анализы методов реализации и построения мобильных корпоративных приложений.

Литература

1. Майорова Е.С., Ошурков В.А., Цуприк Л.С. Современное состояние средств разработки мобильных приложений на платформах iOS, Android и WindowsPhone // Перспективы науки и образования. – 2015. – № 4. – С. 5–8.
2. Танненбаум Э. Компьютерные сети. – 4-е изд. – СПб.: Питер, 2005. – 992 с.
3. Gartner Magic Quadrant for Mobile Application Development Platform [Электронный ресурс]. – Режим доступа: https://www.gartner.com/technology/why_gartner.jsp, своб.
4. Клиент-сервер. Medium [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Клиент_-_сервер, своб.

**Ключков Сергей Витальевич**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4107Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: fluurr@gmail.com

**Государев Илья Борисович**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: goss@corp.ifmo.ru

УДК 004**СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОЛОГИЙ ВЕБ-РАЗРАБОТКИ НА ОСНОВЕ
РАЗЛИЧНЫХ ТЕХНОЛОГИЧЕСКИХ СТЕКОВ****Ключков С.В., 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 исследован вопрос анализа различных технологических стеков, предназначенных для разработки веб-приложений. Рассмотрен общий принцип работы веб-приложения и современные инструменты разработки и взаимодействие между ними, выявлена проблема выбора конкретных технологий из огромного множества существующих на рынке.

Ключевые слова: framework, веб-приложение, веб-разработка, технологический стек, JavaScript, PHP, Python, HTTP, библиотеки.

В настоящее время производится масштабное внедрение интернет-технологий во всех сферах жизни общества, при этом различные области, миграции в веб, такие как экономические, политические и социальные, имеют свой уникальный набор требований, необходимых для успешной интеграции в веб-среду.

Веб-приложение можно разбить на разные категории, каждая из которых является уникальной. В данный момент в области веб-разработки представлены различные языки программирования, типы баз данных (БД), веб-серверы. Все эти инструменты часто объединяют в технологические стеки, предоставляя разработчикам конечный набор инструментов, отлично взаимодействующих между собой.

Современное приложение можно разбить на четыре уровня, три из которых находятся на стороне сервера и еще один на стороне клиента, на рисунке представлена внутренняя структура работы веб-приложения. Клиент может запрашивать простые HTML-файлы или, например, графические файлы (любые статические файлы), в таком случае HTTP-сервер может вернуть эти документы клиенту в виде ответа, не переходя на следующий уровень. Если клиент отправил запрос на файл, который необходимо сгенерировать, то после слоя, на котором находится HTTP-сервер, запрос переходит на следующий уровень – приложение сервера, где в соответствии с запросом генерируется ответ, как правило, HTML-страница.

После чего на уровень клиента отправляется результат через слой HTTP-сервера. В том случае, если требуется записать, удалить, изменить или получить данные из БД, то запрос, дойдя до уровня приложения сервера, интерпретируется для БД и отправляется на следующий уровень – сервер БД, где производится обработка запроса, а затем отправка ответа в обратном направлении клиенту, пройдя необходимую модификацию на всех уровнях на своем пути.

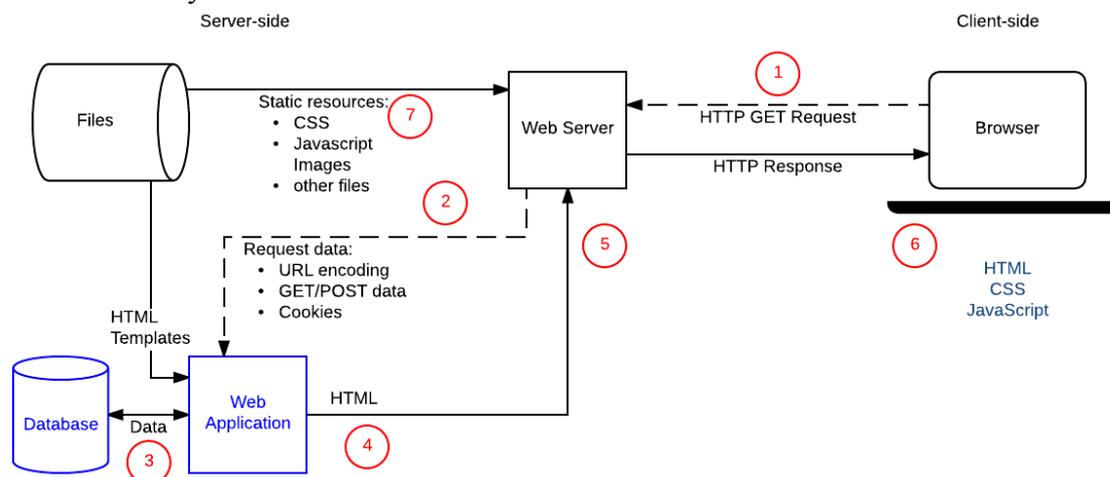


Рисунок. Внутренняя структура веб-приложения

Эти четыре уровня и являются составными компонентами технологических стеков, а точнее инструменты, используемые на каждом из них, рассматриваемые в данной работе.

На стороне клиента происходит реальное взаимодействие с пользователем. Пользователь взаимодействует с веб-сайтом, веб-приложением или мобильным приложением. На этом уровне есть три основные технологии:

1. HTML – язык разметки;
2. CSS – каскадные таблицы стилей;
3. JavaScript – встраиваемый язык для программного доступа к объектам клиентских приложений.

На сегодняшний день существуют три основных фреймворка клиентской разработки на языке JavaScript: AngularJS, ReactJS, VueJS. Все три фреймворка являются независимыми и никак не привязаны к технологиям, используемым на стороне сервера.

Серверная часть приложения включает в себя следующие компоненты: операционная система, HTTP-сервер, серверное приложение, фреймворк. Для написания серверного приложения часто используют следующие языки программирования: PHP, Ruby, Python, Perl, C#, Java, Go, JavaScript (NodeJS).

Такой огромный выбор может сбить с толку. Сложилась такая тенденция, что эти на первый взгляд разрозненные инструменты разработчики формируют в стеки технологий.

Эволюция веб-технологий создает новые условия, в которых современный разработчик должен уметь в течение всей своей профессиональной деятельности ориентироваться в постоянно увеличивающемся объеме технических средств и ресурсов, соответствующих целям разработки и подходящих для конкретных проектов.

Как уже было сказано выше, на сегодняшний день существует огромное количество функциональных средств для разработки веб-приложений. Отдельные языки программирования, каркасы веб-приложений (frameworks), программные библиотеки, операционные системы, веб-серверы, базы данных, объединенные в технологические стеки, имеют конкретную задачу – веб-разработка, но при этом множества получаемых решений все-таки разные.

Любой человек, перед которым стоит задача реализации веб-приложения, будь-то разработчик, заказчик, или обучающийся, сталкивается с проблемой выбора набора

технологий. Но, все же имея большой опыт в разработке, можно добиться объективного выбора.

Все вышесказанное подтверждает недостаточную проработанность в научных исследованиях проблемы выбора технологического стека в веб и определяет актуальность темы данной работы.

Отдельные вопросы выбора технологического стека рассмотрены в работах [1–3]. В исследованиях [4, 5], разбираются методологии проектирования и использования различных технологических стеков. В работах [2, 3] рассмотрены отдельные аспекты проблемы выбора технологического стека.

Литература

1. Громов Ю.Ю., Иванова О.Г., Данилкин С.В. Основы web-инжиниринга: разработка клиентских приложений. – Тамбов: Изд-во ФГБОУ ВПО «ТГТУ», 2012. – 240 с.
2. Полухин П.В. Сравнительный анализ современных веб-технологий [Электронный ресурс]. – Режим доступа: <https://elibrary.ru/item.asp?id=21600980> (дата обращения: 12.12.2017).
3. Гуренко В.В. Сравнительный анализ фреймворков для веб разработки // elibrary.ru. – 2017 [Электронный ресурс]. – Режим доступа: <https://elibrary.ru/item.asp?id=29771187>, своб.
4. Семенов Н. Выбор технологий для большого и не очень большого веб-проекта [Электронный ресурс]. – Режим доступа: https://habrahabr.ru/company/SECL_GROUP/blog/315734/ (дата обращения: 12.12.2017).
5. Веллинг Л. Разработка веб-приложений с помощью PHP и MySQL. – М.: Вильямс, 2016. – 846 с.



Ковалев Василий Васильевич

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4108

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: vasya.kowalev@yandex.ru



Государев Илья Борисович

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: goss@corp.ifmo.ru

УДК 004.75

ОБЗОР МОДЕЛЕЙ ПРЕДОСТАВЛЕНИЯ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ В ВЕБЕ

Ковалев В.В.

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 описаны проблемы вариативности трактовок основных терминов и классификаций в области облачных технологий. Приведена классификация моделей виртуализации ресурсов, моделей обслуживания и развертывания облачных решений. Рассмотрены преимущества и недостатки облачных сервисов в сравнении с простым хостингом.

Ключевые слова: облачные технологии, облака, виртуализация, 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 источников в Интернете показал, что многие аспекты облачных вычислений исследованы довольно подробно, однако встречаются разные трактовки одних и тех же терминов, имеются разные взгляды на классификацию облачных технологий, особенно методов виртуализации.

Целью данной работы являлось решение этой проблемы путем анализа и обобщения источников. В задачи входило определение понятийного ряда в области облачных технологий, выявление преимуществ и недостатков облаков в сравнении с простым хостингом, а также описание существующих классификаций и моделей предоставления облачных вычислений в вебе.

Термин «облачные вычисления» обозначает концепцию предоставления вычислительных ресурсов (от простых приложений до целых центров обработки данных) по требованию через Интернет с оплатой за фактическое использование. При этом осуществляется совместный доступ к общим ресурсам и услугам сервера и обеспечивается их гибкая настройка согласно заданным требованиям.

«Облачные технологии», или «облака» – это сервисы, реализующие данную концепцию. В качестве основы для ее реализации используется идея виртуализации ресурсов.

В результате анализа веб-ресурсов и систематизации информации были определены следующие модели виртуализации (рис. 1, 2):

1. виртуализация приложений (application virtualization);
2. серверная виртуализация (server virtualization):
 - полная виртуализация (full virtualization);
 - паравиртуализация (paravirtualization);
 - аппаратная виртуализация (hardware-assisted virtualization);
 - виртуализация на уровне операционной системы (ОС) (operating-system-level virtualization).



Рис. 1. Схематичное изображение виртуализации приложений (а) и полной виртуализации (б)

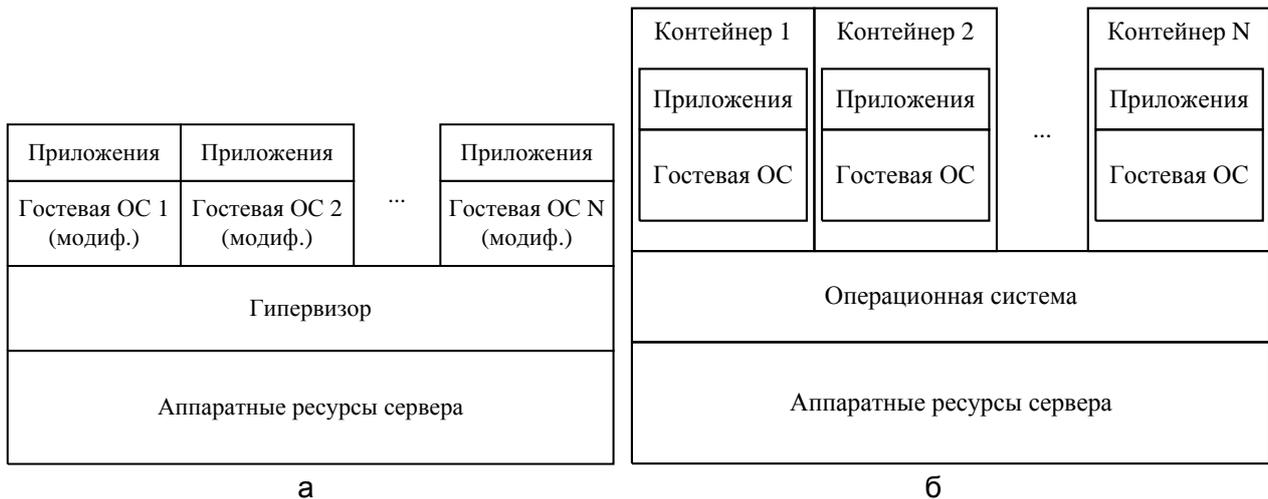


Рис. 2. Схематичное изображение паравиртуализации (а) и виртуализации на уровне операционной системы (б)

При необходимости можно использовать несколько видов виртуализации одновременно и создавать решения любого уровня абстрактности.

Выделяют следующие особенности облаков [1]:

- самообслуживание по требованию. Потребитель в одностороннем порядке может изменять количество используемых ресурсов, таких как серверное время и сетевое

хранилище, при необходимости автоматически без взаимодействия с каждым поставщиком услуг;

- повсеместный доступ. Услуги доступны потребителям по сети передачи данных в независимости от используемого устройства, будь то мобильные телефоны, планшеты, ноутбуки или терминальные станции;
- объединение ресурсов. Ресурсы поставщика услуг (физические и виртуальные) объединены для обслуживания нескольких потребителей и динамически распределяются в зависимости от спроса на вычислительные мощности;
- плата за фактическое использование ресурсов. В отличие от тарифов с жестко заданными параметрами арендуемых ресурсов, в облаках применяется принцип оплаты только реально использованных ресурсов (pay-as-you-go);
- учет статистики. В любой момент можно узнать, сколько именно было задействовано ресурсов. Это может помочь подобрать более подходящий тариф или конфигурацию, а также позволяет делать прогноз пиковых нагрузок в будущем.

Преимущества использования облачных технологий в сравнении с простым хостингом (shared hosting) заключаются в следующем [2]:

- масштабируемость и гибкость настроек. Потребитель в любой момент может изменить конфигурацию для своего облачного решения. Большое количество вариантов настроек позволяет подобрать оптимальное соотношение производительности и цены;
- неограниченное количество сайтов, доменов и аккаунтов. Это достигается за счет виртуализации;
- повышенный уровень безопасности. Так как все потребители изолированы друг от друга, то взлом одного не скажется на безопасности других.

Недостатками использования облачных технологий являются:

- единая точка отказа. При отказе физического сервера, все виртуальные серверы, развернутые на нем, также станут недоступны. Выходом из этой ситуации является использование нескольких объединенных серверов (server farms, серверные фермы) или использования мультиоблака;
- ограниченность уровней виртуализации. Большое количество уровней виртуализации может замедлить работу сервера;
- высокая цена ошибок. В результате того, что облака становятся все более популярными, любые уязвимости в безопасности (например, настройка прав доступа, пароли администраторов и так далее) могут привести к утечке информации огромного количества пользователей.

На сегодняшний день существует несколько моделей облачных вычислений. Они различаются количеством потребителей, делящих между собой ресурсы сервера, количеством ресурсов, которые может использовать каждый из потребителей, тарифом и используемыми технологиями виртуализации. Ниже будут рассмотрены модели обслуживания облачных решений и модели их развертывания.

Модели обслуживания (service models) различаются по предоставляемому потребителю ресурсу или комплекту ресурсов. Существует три основных модели обслуживания облачных решений [3]:

1. программное обеспечение как услуга (software as a service, SaaS);
2. платформа как услуга (platform as a service, PaaS);
3. инфраструктура как услуга (infrastructure as a service, IaaS).

На рис. 3 схематично показано, какие возможности доступны при использовании каждой модели обслуживания

Модели развертывания (deployment models) облачных решений различаются по количеству потребителей на одном физическом сервере и методу предоставления пространства на нем. Выделяют четыре основных модели развертывания [4]:

1. частное облако (private cloud);

2. общественное облако (community cloud);
3. публичное облако (public cloud);
4. гибридное облако (hybrid cloud).

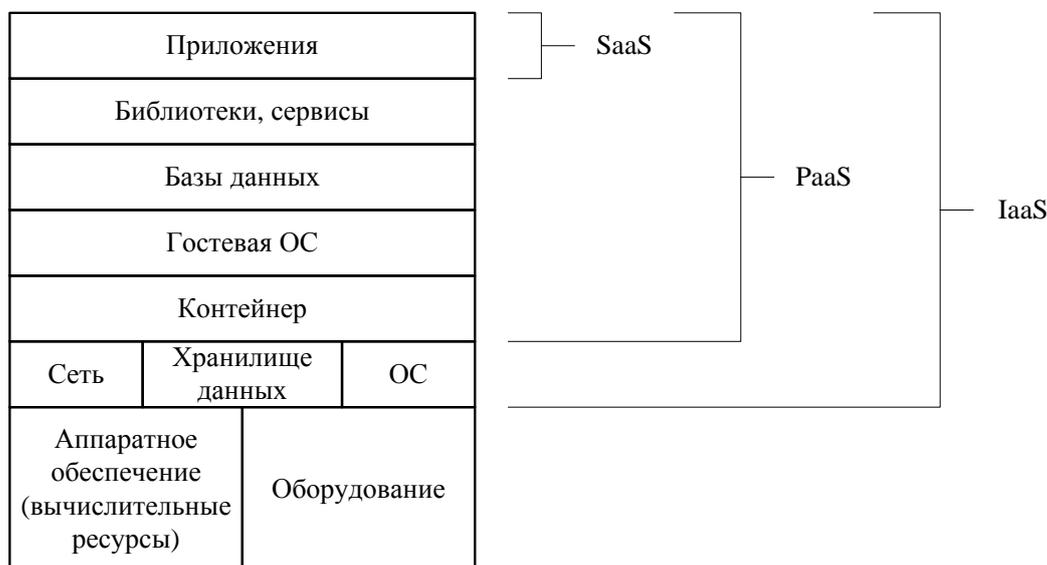


Рис. 3. Возможности по управлению вычислительными ресурсами для каждой модели обслуживания

В результате исследования был определен понятийный ряд в области облачных технологий, выявлены преимущества и недостатки облаков в сравнении с простым хостингом, а также описаны существующие классификации и модели предоставления облачных вычислений в вебе [5, 6].

Литература

1. Довгаль В.А. Облачные вычисления и анализ вопросов информационной безопасности в облаке // Вестник Адыгейского государственного университета. – 2015. – № 2(161). – С. 159–166.
2. Что такое VPS хостинг? [Электронный ресурс]. – Режим доступа: <https://adminvps.ru/blog/what-is-vps-hosting/> (дата обращения: 28.02.2018).
3. Chen Y., Li X., Chen F. Overview and Analysis of Cloud Computing Research and Application // International Conference on E-Business and E-Government (ICEE). – 2011. – P. 1–4.
4. X-as-a-services: как не погрязнуть в аббревиатурах облачных услуг [Электронный ресурс]. – Режим доступа: <http://iaas-blog.it-grad.ru/tendencii/x-as-a-services-kak-ne-pogryaznut-v-abbreviaturax-oblachnyx-uslug/> (дата обращения: 28.02.2018).
5. The NIST Definition of Cloud Computing [Электронный ресурс]. – Режим доступа: <http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf> (дата обращения: 28.02.2018).
6. Why you should care about multicloud [Электронный ресурс]. – Режим доступа: <https://www.infoworld.com/article/2611544/hybrid-cloud/why-you-should-care-about-multicloud.html> (дата обращения: 28.02.2018).



Ковеченков Владислав Олегович

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: vladislav.kovechenkov@gmail.com

УДК 004.77

**ИССЛЕДОВАНИЕ ИМПЛЕМЕНТАЦИИ МЕТОДОВ
МАШИННОГО ОБУЧЕНИЯ В ВЕБ**

Ковеченков В.О.

Научный руководитель – к.ф.-м.н. Соловьев И.О.

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

В работе рассмотрены перспективы разработки веб-приложений, использующих методы машинного обучения в экосистеме языка программирования JavaScript с целью снижения временных и ресурсных затрат по имплементации методов машинного обучения в рамках веб-приложения. Приведены существующие инструменты имплементации методов машинного обучения, а также технологии, на которых они основаны. Дано детальное обоснование потенциалу применения JavaScript для разработки веб-приложений, связанных с имплементацией методов машинного обучения.

Ключевые слова: машинное обучение, веб-приложение, JavaScript, WebAssembly, нейронные сети, браузер.

Современные кампании и некоммерческие организации практически в любой сфере сталкиваются с необходимостью обрабатывать огромные массивы данных. Для обработки данных могут использоваться методы машинного обучения, с помощью которых решаются различные задачи. Например, анализ портрета клиента позволяет предоставить отдельному пользователю персонализированное обслуживание; анализ локации, типа устройства пользователя и истории посещения веб-сайтов позволяют определить, насколько эффективно предоставление конкретной рекламы определенной группе пользователей.

На момент проведения исследования одним из доминирующих языков имплементации методов машинного обучения являлся Python [1]. Это обуславливается множеством причин, самыми важными из которых являются: обширное количество библиотек, активная поддержка развивающегося сообщества и производительность. Многие Python-библиотеки обращаются к модулям, написанным на языке C или Fortran, тем самым используя преимущества в производительности низкоуровневых языков программирования. Также некоторые библиотеки используют вычислительные мощности графических процессоров, что также значительно повышает эффективность вычислений. Некоторые библиотеки при этом позволяют осуществлять подход «одиночный поток команд, множественный поток данных» (ОКМД). В таком случае матричные и векторные вычисления имеют сложность $O(1)$. В реализации методов машинного обучения этот момент является критическим, так как подобные операции выполняются очень часто в ходе проведения анализа [2].

Однако если рассматривать разработку веб-приложений, то JavaScript применяется для решения огромного количества задач. Это стремительно развивающийся язык программирования, обширное сообщество которого находит новые способы применения для JavaScript в различных сферах, не только в разработке клиентской части веб-приложений.

Разработчики со всего мира предоставляют сообществу новые библиотеки и фреймворки, тем самым расширяя возможности JavaScript. С развитием и увеличением популярности JavaScript, имплементация машинного обучения в веб может позволить применять к некоторым задач концепцию моноязыковой разработки – когда разработка ведется в рамках одного языка программирования. Некоторые компании и некоммерческие организации в своих проектах внедряют концепцию моноязыковой разработки с целью облегчения коммуникации разработчиков.

Для реализации методов машинного обучения в экосистеме JavaScript уже существуют некоторые инструменты. Некоторые из них позволяют снизить требования к серверу, который обеспечивает работу веб-приложения, перенося вычисления, необходимые для реализуемой модели машинного обучения, на сторону клиента. Одним из таких инструментов является фреймворк TensorFire, позволяющий запуск нейронных сетей в браузере при поддержке технологии WebGL.

Keras.js позволяет запускать модели, которые были подготовлены с помощью различных инструментов, в том числе библиотек Python [3]. Модели могут быть запущены в браузере при поддержке WebGL, а также в среде Node.js.

Deeplearn.js библиотека для машинного обучения также работающая в браузере. Deeplearn.js дает возможность обучать нейронные сети в браузере, избегая установки дополнительного софта или бэкенда. Исследователи отметили, что клиентская библиотека для машинного обучения может использоваться для интерактивных презентаций, быстрого прототипирования и визуализации и даже офлайн-вычислений. Кроме того, браузер – это одна из самых популярных платформ для разработки.

Основной проблемой, как упоминалось выше, является то, что JavaScript – однопоточный язык программирования. Однако развитие технологии WebAssembly позволит JavaScript работать в связке с низкоуровневыми языками программирования, делегируя многопоточные вычисления модулям, реализованным с помощью других технологий.

Таким образом, следует отметить, что на момент реализации в современных браузерах технологии WebAssembly разработчики смогут применять новые методы имплементации методов машинного обучения в браузере.

Литература

1. Witten I.H., Frank E., Hall M.A. Data Mining: Practical Machine Learning Tools and Techniques. – Morgan Kaufmann Publishers Inc. San Francisco. – 2011. – 630 p.
2. Isoni A. Machine Learning for the Web. – Packt Publishing, 2016. – 298 p.
3. Raschka S., Mirjalili V. Python Machine Learning. – 2nd Edition. – Packt Publishing, 2017. – 622 p.



Козак Олег Олегович

Год рождения: 1990

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, аспирант

Направление подготовки: 09.06.01 – Информатика и вычислительная
техника

e-mail: lega152@yandex.ru



Зими́на Дина Викторовна

Год рождения: 1984

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, аспирант

Направление подготовки: 09.06.01 – Информатика и вычислительная
техника

e-mail: dinazi@mail.ru

УДК 004.588

МЕТОДЫ И СРЕДСТВА МОДЕЛИРОВАНИЯ ТЕХНОЛОГИЙ ДИСТАНЦИОННОГО ОБУЧЕНИЯ

Козак О.О., Зими́на Д.В.

Научный руководитель – д.т.н., профессор Погорелов В.И.

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

В работе рассмотрены различия в классическом очном и дистанционном обучении. Дано определение понятию e-learning, возможное его внедрение в качестве одного из составляющих смешанного обучения. Рассмотрены достоинства и недостатки разных типов платформ для разработки онлайн-курсов. Выявлена проблема выбора системы управления обучением для проектирования новых экспериментальных элементов. Предложен способ ее решения с помощью использования протокола LTI.

Ключевые слова: дистанционное обучение, учебный процесс, информационные технологии, Open Source, e-learning.

Цель работы – обзор достоинств и недостатков онлайн-обучения, а также методов и средств для создания онлайн-курсов.

В классической форме очного обучения материал можно разделить условно на теоретический и практический. Теоретический материал подается на лекциях; полученные навыки можно затем отработать и закрепить на практических, лабораторных и семинарских занятиях. При дистанционной форме обучения отсутствует личный контакт преподавателя и учащегося. В этом заключается главная сложность в управлении учебным процессом. Необходимость заниматься самостоятельно накладывает дополнительную ответственность на учащегося, а полноценный контроль его знаний, умений и навыков заметно усложняется. Современным ответом на озвученные сложности является электронное обучение, или e-learning. E-learning – одна из главных тенденций развития современного образования. Данная форма обучения имеет множество плюсов; традиционные формы подачи материала и контроля обучения в ней автоматизированы. Электронное обучение может быть как самостоятельным инструментом самообразования, так и входить в состав традиционного образовательного процесса. Информационные технологии берут на себя выполнение рутинных функций, а также делают процесс обучения более доступным большему числу желающих.

Отдельно стоит сказать о смешанном обучении. Как следует из названия, смешанное обучение предполагает сочетание традиционной очной формы с технологиями

дистанционного обучения. В этом случае информационные технологии отвечают за представление учебного материала, тогда как контроль знаний и организационные вопросы решает лично преподаватель.

Самой серьезной является проблема выбора платформы для разработки курса онлайн-обучения. Самым очевидным является выбор в пользу платного программного обеспечения. Это коммерческие продукты с хорошей технической поддержкой и регулярными обновлениями. Однако необходимость оплачивать продукт сама по себе является серьезным недостатком таких систем. Причем оплата может взиматься как разово, так и по подписке, а также за дополнительных пользователей.

Можно также использовать в качестве средства для разработки курсов дистанционного обучения одну из систем на базе Open Source (OS) решений. Наиболее популярны в данном направлении системы Moodle и Sakai. Кроме того, некоторые платформы, например, ресурс Stepik [1], предоставляют возможность самостоятельного и бесплатного конструирования и администрирования онлайн-курсов на своей платформе.

Преобразование учебного курса в онлайн-формат можно свести к следующей последовательности шагов:

1. разработка модели будущего онлайн-курса, его общей схемы;
2. выбор системы для размещения онлайн-курса;
3. разделение существующего материала по формам и методам обучения;
4. определение онлайн-аналогов для существующих форм обучения;
5. создание фонда оценочных средств;
6. перевод учебного материала в электронный вид согласно инструментам выбранной системы;
7. настройка сроков и других параметров онлайн-курса.

Стоит отметить, что при самостоятельной разработке новых элементов в онлайн-обучение возникает дилемма. С одной стороны, готовые платформы для разработки онлайн-курсов не предоставляют большой свободы для доработки собственных дополнений и плагинов. С другой, создание своей системы управления обучением с нуля чрезвычайно трудоемки. Выходом из сложившейся ситуации может стать применение протокола LTI (Learning Tools Interoperability) [2]. Этот протокол позволяет подключить к одной системе обучения элементы функционала другой – таким образом, можно разработать собственный функционал, дополнив его готовыми структурными элементами из внешней системы. Использование протокола LTI доступно для таких систем как Coursera, Moodle и Stepik [3].

Выводы:

1. основной недостаток дистанционного обучения – трудности в контроле результатов обучения – можно решить с помощью смешанного и онлайн-обучения;
2. e-learning, или электронное обучение, автоматизирует часть рутинных задач традиционной формы обучения;
3. для самостоятельной разработки новых элементов онлайн-обучения можно воспользоваться сочетанием существующих платформ и систем на базе Open Source решений.

Литература

1. Stepik – Бесплатные онлайн-курсы [Электронный ресурс]. – Режим доступа: <http://stepik.org/>, своб.
2. Learning Tools Interoperability | IMS Global Learning Consortium [Электронный ресурс]. – Режим доступа: <https://www.imsglobal.org/activity/learning-tools-interoperability>, своб.
3. Андреев А.М. Выбор системы управления обучением для высшего учебного заведения // Новая наука: стратегии и векторы развития. – 2016. – № 3-1. – С. 3–6.



Костюченко Юлия Андреевна

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: jukostyuchenko@gmail.com

УДК 004

АНАЛИЗ ПОДХОДОВ К МОДЕЛИРОВАНИЮ ДАННЫХ С ПОМОЩЬЮ БИБЛИОТЕК ЯЗЫКА PYTHON

Костюченко Ю.А.

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 подходы к моделированию данных с помощью библиотек языка Python, предназначенных для обработки больших объемов данных и машинного обучения. Выделены особенности и проанализированы различия нескольких библиотек, входящих в состав набора SciPy Stack.

Ключевые слова: моделирование данных, Python, машинное обучение, анализ данных.

Машинное обучение стало неотъемлемой частью различных коммерческих и исследовательских проектов, начиная с постановки медицинского диагноза с последующим лечением и заканчивая поиском друзей в социальных сетях. На таких популярных сайтах, как Facebook, Amazon и Netflix, каждый раздел сайта содержит несколько моделей машинного обучения.

Наиболее успешные алгоритмы машинного обучения – те, которые автоматизируют процессы принятия решений путем обобщения известных примеров. В этих методах, известных как обучение с учителем или контролируемое обучение (supervised learning), пользователь предоставляет алгоритму пары объект-ответ, а алгоритм находит способ получения ответа по объекту. В частности, алгоритм способен выдать ответ для объекта, которого он никогда не видел раньше, без какой-либо помощи человека. Примером подобной задачи может быть определение почтового индекса по рукописным шрифтам на конверте. Здесь объектом будет сканированное изображение почерка, а ответом – фактические цифры почтового индекса. Чтобы создать набор данных для построения модели машинного обучения, необходимо собрать большое количество конвертов. Затем можно самостоятельно прочитать почтовые индексы и сохранить цифры в виде ответов.

С момента своего появления в 1991 году, Python стал одним из самых популярных динамических языков программирования, наряду с Perl, Ruby и другими. В последние годы Python приобрел большую популярность в области машинного обучения и Data Science. Он сочетает в себе мощь и простоту использования предметно-ориентированных скриптовых языков типа MATLAB и R. В Python есть библиотеки для анализа данных, визуализации, статистических вычислений, обработки естественного языка, обработки изображений и многого другого.

Рассмотрим основные библиотеки, используемые для машинного обучения на языке Python, некоторые из которых образуют так называемый SciPy Stack. Python SciPy Stack – набор библиотек, специально предназначенных для научных вычислений. Каждый, кто собирается использовать Python в науке, должен познакомиться с этим стеком.

1. Самый фундаментальный пакет – NumPy. Он позволяет выполнять основные операции над n -массивами и матрицами: сложение, вычитание, деление, умножение, транспонирование, вычисление определителя и т.д. Благодаря механизму векторизации, NumPy повышает производительность и, соответственно, ускоряет выполнение операций [1].

NumPy – один из основных пакетов для научных вычислений. Он содержит функциональные возможности для работы с многомерными массивами, высокоуровневыми математическими функциями (операции линейной алгебры, преобразование Фурье, генератор псевдослучайных чисел). Массив NumPy является основной структурой данных. Базовый функционал NumPy – это класс `ndarray`, многомерный массив. Всего его элементы должны быть одного и того же типа (рис. 1, а).

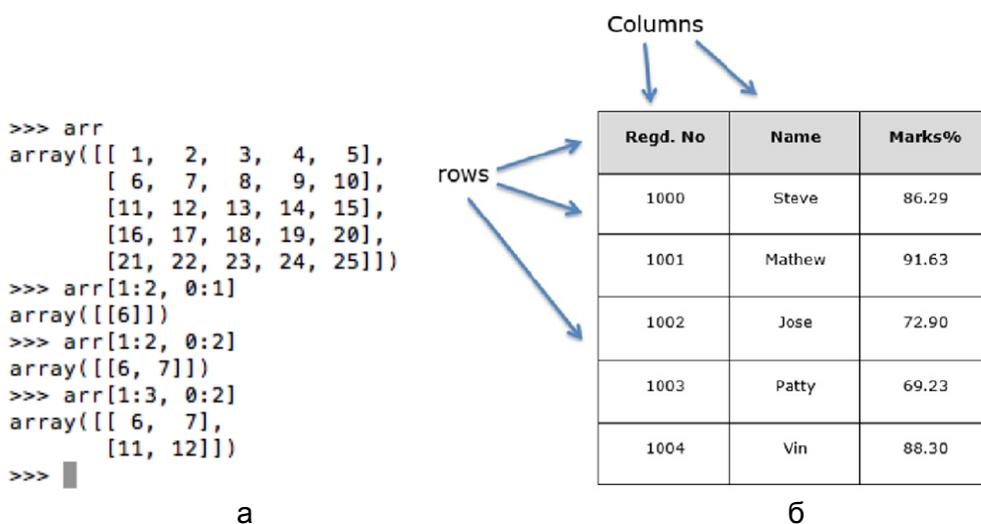


Рис. 1. Массив данных в NumPy (а) и Pandas DataFrame (б)

2. Pandas – это пакет, предназначенный для простой и интуитивно понятной работы с «помеченными» и «реляционными» данными. Тоже работает в связке с NumPy, и помимо математических вычислений обеспечивает их агрегацию и визуализацию.

Pandas пакет Python обеспечивает быстрые, гибкие и выразительные структуры данных, предназначенные для упрощения работы с «реляционными» или «маркированными» данными. Он стремится быть основным высокоуровневым блоком для выполнения анализа данных в режиме реального времени на языке Python. Кроме того, он имеет цель стать самым мощным и гибким инструментом с открытым исходным кодом, инструментом для анализа данных и манипуляции с ними, доступным на любом языке. Pandas хорошо подходит для различных видов данных:

- табличные данные, такие как в таблице SQL или Excel;
- упорядоченные и неупорядоченные данные (не обязательно с фиксированной частотой) временных рядов;
- произвольные матричные данные с метками строк и столбцов;
- любая другая форма наблюдений и статистические наборы данных. Данные, фактически не требующие наличия меток.

Одной из основных структур Pandas является объект DataFrame. Данный объект лучше всего представлять себе в виде обычной таблицы и это правильно, ведь DataFrame является табличной структурой данных [2]. В любой таблице всегда присутствуют строки и столбцы. Столбцами в объекте DataFrame выступают объекты Series, строки которых являются их непосредственными элементами (рис. 1, б).

3. Matplotlib за более чем 10 лет своего существования уже фактически стала стандартом визуализации на Python. Многие современные библиотеки для построения графиков проектируются для работы совместно с matplotlib. Некоторые библиотеки, например,

pandas или Seaborn, представляют из себя обертки над matplotlib. Однако вместе с широтой возможностей приходит и сложность в устройстве, и, как следствие, работать с библиотекой тоже не всегда легко. Еще один минус – оформление в духе 1990-х годов, которое явно не подойдет для презентаций. Последнее, однако, должен решить релиз 2.0.

С помощью matplotlib можно создать:

- линейные графики;
- графики рассеяния;
- гистограммы;
- круговые диаграммы;
- стволые участки;
- контурные участки;
- участки колчана;
- спектрограммы [3].

Библиотека низкоуровневая, что означает большой объем кода для расширенной визуализации. Но производительность и работа с привычным языком позволяют закрыть глаза на этот недостаток.

Matplotlib является основным инструментом для визуализации данных на языке Python, поддерживается различными платформами и IDE (iPython, Jupyter и пр.).

4. SciPy – небольшая библиотека, предназначенная для выполнения научных и инженерных расчетов. Содержит в себе модули для линейной алгебры, оптимизации, интеграции и статистики (рис. 2).

Основные возможности:

- поиск минимумов и максимумов функций;
- вычисление интегралов функций;
- поддержка специальных функций;
- обработка сигналов;
- обработка изображений;
- работа с генетическими алгоритмами;
- решение дифференциальных уравнений.

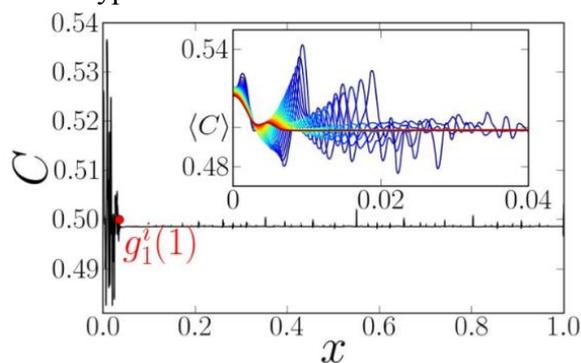


Рис. 2. Визуализация данных с помощью SciPy

5. Theano – одна из самых мощных библиотек в нашем списке. Вот несколько причин:

- тесная интеграция с NumPy;
- использование CPU и GPU для повышения производительности;
- встроенные механизмы оптимизации кода;
- расширения для юнит-тестирования и самопроверки.

Theano используется там, где необходимо произвести вычисления с большой точностью максимально быстро, например, в нейронных сетях и машинном обучении.

6. TensorFlow. Библиотека от Google была разработана специально для обучения нейронных сетей. Вы можете прямо сейчас оценить эффективность TensorFlow на примере голосового помощника или в игре «Quick, Draw». Библиотека использует многоуровневую

систему узлов для обработки большого количества данных, что расширяет сферу ее использования далеко за научную область.

7. IPython. IPython – это командная оболочка для интерактивных вычислений на нескольких языках программирования, изначально разработана для языка программирования Python, что предлагает расширенные возможности, мультимедиа, дополнительный синтаксис оболочки, автодополнение. IPython в настоящее время предоставляет следующие возможности:

- мощные интерактивные оболочки (терминал на основе Qt);
- редактор на базе браузера с поддержкой кода, текста, математических выражений и др.;
- поддержка интерактивной визуализации данных и использования GUI;
- легкий в использовании, высококачественные инструменты производительности для параллельных вычислений.

Это лишь малая часть Python-библиотек, используемых в Data Science. Некоторые из них приносят общую практическую пользу, другие предлагают оптимизацию под конкретные задачи.

Литература

1. Маккинни У. Python и анализ данных. – М.: ДМК Пресс, 2015. – 482 с.
2. Рашка С. Python и машинное обучение. Packt Publishing. – М.: ДМК Пресс, 2017. – 418 с.
3. Matplotlib [Электронный ресурс]. – Режим доступа: <https://matplotlib.org/>, своб.



Кузнецова Наталья Валерьевна

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: nvkuznetsova@list.ru

УДК 004.4

ОБЗОР МЕТОДИК РАЗРАБОТКИ НА PHP

Кузнецова Н.В.

Научный руководитель – к.ф.-м.н. Соловьев И.А.

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

В работе рассмотрены методики разработки на платформе PHP, их преимущества. Представлены результаты сравнения методик, сделаны выводы об областях использования конкретной методики.

Ключевые слова: методика разработки, PHP, CMS, фреймворк.

Интернет – важный источник информации, который ежедневно пополняется новыми сайтами и приложениями. Получая заказ на создание веб-приложения, программист должен определить подход к разработке, учитывая функциональные требования приложения, выделенный бюджет, специфичность функционала и другие требования. Если у опытного программиста данный вопрос вызывает меньше затруднений, то у менее опытного специалиста могут возникнуть сомнения относительно используемой технологии.

Веб-приложения строятся на основе клиент-серверной архитектуры. Наиболее распространенной платформой разработки серверной части веб-приложения является язык PHP. Причины широкого использования данной технологии в веб-разработке – простота языка по сравнению с аналогами, поддержка большого количества функций и библиотек, простота работы с различными базами данных, поддержка большинством веб-серверов [1].

Разработать веб-приложение можно тремя способами: самостоятельно написать код с нуля, воспользоваться системой управления контентом, использовать для разработки фреймворк. Первый способ подойдет для небольших несложных приложений. Когда речь заходит о разработке крупного приложения с большим функционалом данный подход приведет к появлению ошибок и увеличению времени разработки из-за большого количества рутинных операций, с которыми придется столкнуться разработчикам.

Для уменьшения ошибок в коде и сокращения времени разработки нужно использовать систему управления контентом или фреймворк. Веб-фреймворки и CMS – популярные средства создания веб-приложений. Большинство веб-фреймворков и CMS имеют в своей основе именно PHP.

CMS – это информационная система или программное решение, которое позволяет быстро создавать шаблонные сайты [2]. Для использования CMS нет необходимости обладать глубокими знаниями в веб-разработке, что является главным преимуществом данного подхода. CMS предоставляет пользователю готовый интерфейс, с помощью которого можно редактировать содержимое сайта, наполнять его контентом и в короткие сроки предоставлять готовый результат. Многие CMS также позволяют работать с данными, автоматизируя процессы работы с базой данных [2]. Таким образом, CMS – удобное средство разработки, которое подойдет как новичкам в области веб-разработки, так и

профессионалам. Наиболее популярные системы управления контентом: 1С-Битрикс, WordPress, Dupal, Joomla!

Фреймворк позволяет разработчику не начинать работу с нуля, а предоставляет готовую структуру приложения, которую можно наполнять необходимым функционалом и дополнительными модулями. Веб-фреймворк позволяет сконцентрироваться на реализации более специфических функций приложения без необходимости выполнять рутинные действия по созданию каркаса приложения.

В основе фреймворков обычно лежит архитектура MVC. Шаблон MVC позволяет разделить бизнес-логику приложения и интерфейса пользователя [1]. Таким образом, приложение разделяется на компоненты так, чтобы изменения одного компонента не затрагивали другие. Данная архитектура предлагает модульный подход к написанию приложения, снижает сложность архитектуры приложения, повышает гибкость приложения и возможность повторного использования кода. Преимущества паттерна MVC позволяют создать хороший каркас будущего веб-приложения.

Использование веб-фреймворка целесообразно в следующих ситуациях [3]:

1. при проектировании таких стандартных приложений как интернет-магазины, социальные сети, новостные порталы и др.;
2. когда необходимо обеспечить масштабируемость приложения по мере его роста;
3. если необходимо создать модульное приложение с максимально возможным повторным использованием кода.

Фреймворки также незаменимы при командной разработке. Готовый каркас приложения определяет стандарт, в рамках которого должны работать программисты. Наиболее часто в проектах используют фреймворки: Laravel, CodeIgniter, Symfony, Zend, Yii.

В таблице приведено сравнение методик разработки по некоторым критериям.

Таблица. Сравнение методик разработки

	Написание с нуля	CMS	Фреймворк
Время разработки	Долго	Быстро	Зависит от приложения
Вероятность ошибок	Высокая	Снижена	Снижена
Масштаб проекта	Небольшие	Небольшие – крупные	Небольшие – крупные
Скорость выполнения запросов	Высокая	Обычно ниже, чем при написании с нуля	Обычно ниже, чем при написании с нуля
Потребление памяти	Небольшое	Больше, чем при написании с нуля	Больше, чем при написании с нуля

Для готового приложения особенно важны скорость выполнения запросов и потребление памяти. Однако, несмотря на то, что CMS и фреймворки уступают по этим критериям чистому коду, не следует забывать о других важных для разработки критериях и проблемах, для решения которых были созданы данные технологии. Выбирая способ разработки, нужно опираться на требования заказчика и известные данные о производительности нескольких технологий, чтобы предложить наиболее подходящий вариант.

На выбор методологии и технологии разработки влияет комплекс различных факторов: требования заказчика, время, необходимое для разработки продукта, масштаб проекта и другие факторы. Если масштабы проекта небольшие или функционал приложения отличается сложностью, то можно разрабатывать с нуля, а если необходимо предоставить результат в очень сжатые сроки, то, имеет смысл использовать CMS. Если проект небольшого масштаба имеет достаточно специфический функционал, то нужно использовать один из фреймворков, например, CodeIgniter. Для средних и крупных долгосрочных проектов подойдут такие фреймворки как Laravel и Symfony.

Литература

1. Лавлинская О.Ю., Бондаренко М.С. Обзор инструментальных средств и технологий разработки веб-приложений // Вестник Воронежского института высоких технологий. – 2016. – № 4 – С. 73–78.
2. Степанов А.Б. Анализ целесообразности применения фреймворков и CMS при разработке сайта или веб-приложения // Университетская наука. – 2017. – Т. 2. – С. 175.
3. Bartosz P., Karol P., Nowak L. Building PHP Applications with Symfony, CakePHP, and Zend Framework. – John Wiley and Sons, 2011. – 528 p.

**Линькова Татьяна Игоревна**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4105Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: tanyakalypso@gmail.com

**Шуклин Дмитрий Анатольевич**

Год рождения: 1979

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: do@limtu.ru

УДК 003

**МЕТОДИКИ ТИПОГРАФИКИ В ДИЗАЙНЕ ПОЛЬЗОВАТЕЛЬСКИХ
ИНТЕРФЕЙСОВ ВЕБ-САЙТОВ****Линькова Т.И., Шуклин Д.А.****Научный руководитель – к.педагог.н., доцент Шуклин Д.А.**

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

В работе рассмотрена роль типографики в современном веб-дизайне и дизайне пользовательских интерфейсов, а также приемы и методики микро- и макро-типографики, используемых для веб-сайтов.

Ключевые слова: типографика, веб-дизайн, пользовательский интерфейс, текст, macro-typography, micro-typography, readability, legibility.

Так как по-прежнему большая часть информации в Интернете является текстовой, современные веб-дизайнеры и UI-дизайнеры просто обязаны уделять должное внимание тексту, его корректности, читабельности, а с развитием мобильных устройств еще и адаптивности. Чтобы создать эффективный пользовательский интерфейс и формат контента, который будет удобен для пользователя, дизайнер непременно должен изучить основы типографики.

Эффективная типографика является ключом к мощному дизайну. Его эффективность зависит не только от его содержания, но и от презентации. Дизайнеры могут передавать определенное настроение или сообщение, выбирая подходящие шрифты и способы их представления. Таким образом, типографика помогает дизайну общаться с людьми. Типографика существенно влияет на первое впечатление о программном продукте или веб-странице.

В работах зарубежных ученых достаточно много информации по данной тематике, однако среда развития этой проблемы очень динамична, поэтому исследования в этой теме всегда будут востребованы.

Как отметил Оливер Риченштейн еще в 2006 году: «Веб-дизайн – это на 95% типографика». И действительно, 95% информации в Интернете – это письменный язык [1]. Помочь выделить из огромного потока информации в Интернете нужную, сделать эту

информацию доступной, удобочитаемой и как можно более понятной – задачи для информационных дизайнеров, дизайнеров интерфейсов и веб-дизайнеров. Именно поэтому как разработчику, так и дизайнеру интерфейсов, так важно понимать все аспекты типографики применительно к веб-сайтам. Также стоит отметить что типографика разделяется на макро- и микротипграфику. Макротипографика в основном ориентирована на подбор размера и цвета шрифта для улучшения читаемости, а также определяет расстояния в макете, иерархию и ритм. Иначе говоря, отвечает за общую структуру текстовой части в контексте дизайна сайта в целом. Микротипграфика фокусируется на разборчивости, т.е. рассматривает такие вещи, как форму самой буквы, символы и расстояние между ними [2].

Макротипографика (общая текстовая структура) в отличие от микротипграфики (подробные аспекты букв и интервалов между ними) охватывает многие аспекты того, что в настоящее время называется «информационным дизайном» [1]. Создание текста, легкого для понимания пользователем, возможно только при тщательном планировании типографики и удобочитаемости с учетом всех влияющих на нее факторов. Тремя главными составляющими работы с цифровой информацией являются контент, контекст и пользователь. В ходе анализа первой составляющей, а именно контента, Марко Дугонич [3] предлагает разработчикам создать рабочие файлы HTML и CSS и поместить в них все варианты начертания шрифта и стилевые правила, которые послужат основой типографического стиля проекта. Некоторые исследователи при такой оценке контента также предлагают вести и оценку читабельности с помощью индекса удобочитаемости Флеша, который предлагается высчитывать с помощью специальных скриптов [3]. Такая оценка позволяет с большей рациональностью определиться с величиной отступов в абзацах, задать правильный ритм и темп чтения.

Необходимо также учитывать пользователя и его особенности, тем более если дело касается многоязычного сайта, где на кнопке пространство рассчитано ровно для надписи «Return» на английском, тогда как, например, для «Povratak» на хорватском его может не хватить [4]. Также остро стоит проблема читабельности шрифтов для людей с ограниченными возможностями, например – больных дислексией (рис. 1) или людей с нарушенным цветовым зрением.



hnu oaec bp qd
hnu oaec bp qd

Рис. 1. Шрифт сверху – Futura; шрифт снизу – Open Dyslexic

Выбор гарнитуры шрифта – масштабная задача макротипографики. Наличие или отсутствие засечек особенно не влияет на читабельность, особенно в нынешнюю эпоху экранов с высокой пиксельной плотностью. Однако именно гарнитура задает тон всему веб-сайту, неправильный выбор может исказить замысел или разрушить желаемую атмосферу сайта.

Все чаще можно встретить доводы в пользу продуманного комбинирования шрифтов с засечками и без засечек для улучшения удобочитаемости и эстетической ценности контента. Дуглас Бонэвил [5] рассматривает преимущества и позитивный опыт комбинирования шрифтов с засечками и без засечек. Особенно применяется противопоставление шрифта с засечками в основном тексте и шрифта без засечек в заголовке (или наоборот).

Также можно выделить ряд рекомендаций по макротипографике для веб-дизайнеров:

– хорошая строка текста содержит 40–90 символов, слишком длинные строки неудобно читать, внимание пользователя рассеивается или переключается на другие вещи. Если

- строка слишком короткая, то взгляду слишком часто приходится возвращаться к началу строки, что сбивает правильный темп чтения и не способствует усваиванию информации;
- для того, чтобы текст читался легко, абзацы необходимо разделять, это можно сделать двумя способами: «швейцарский» способ и книжный способ. Книжный способ использует для разделения красную строку – небольшой сдвиг первой строки каждого нового абзаца, швейцарский же способ подразумевает разделение абзацев пустыми строками между ними. Не стоит использовать два этих способа одновременно или комбинировать их;
 - выравнивать текст в абзаце стоит преимущественно по левому краю, так текст выглядит для пользователя наиболее привычным образом и легко читается. Выравнивание по ширине также можно иногда использовать, но чаще всего при таком выравнивании в тексте остаются неравномерные пробелы, которые портят общий вид;
 - заканчиваться строка не должна союзом или предлогом;
 - для создания небольшого аккуратного блока текста лучше всего применить так называемую «висячую» пунктуацию;
 - тенденцией для использования в веб-дизайне на данный момент считается «журнальная» верстка, например, в несколько колонок.

Многие исследователи сходятся во мнении, что в веб-дизайне остро стоит вопрос хинтинга веб-шрифтов, что очень отражается на разборчивости текста. На компьютерах с системой Windows, которая использует механизм визуализации ClearType, отлично выглядят шрифты с качественным хинтингом. Однако хинтинг (или экранная оптимизация) – процесс приспособления шрифтов для максимальной читабельности на экране компьютера требует большого количества времени, поэтому качество далеко не всех гарнитур для веба доведено до идеала. На рисунке показаны скриншоты шрифтов Verdana, Proxima Nova, Arial, Adelle Sans и Georgia, кегль 16 пикселей (кроме Proxima – 18 пикселей), сделанные в Firefox 18 на Windows 7. На рис. 2 Proxima и Adelle рендерятся как веб-шрифты, остальные встроены в большинство операционных систем. Для скриншота верхних 5 строчек был предварительно отключен ClearType.

Gojazni đaćić s biciklom drži hmelj i finu vatu u dže
Gojazni đaćić s biciklom drži hmelj i finu vatu u džepu
Gojazni đaćić s biciklom drži hmelj i finu vatu u džepu nošnje.
Gojazni đaćić s biciklom drži hmelj i finu vatu u džepu nošnje
Gojazni đaćić s biciklom drži hmelj i finu vatu u džepu nošnje.

Gojazni đaćić s biciklom drži hmelj i finu vatu u dže
Gojazni đaćić s biciklom drži hmelj i finu vatu u džepu
Gojazni đaćić s biciklom drži hmelj i finu vatu u džepu nošnje.
Gojazni đaćić s biciklom drži hmelj i finu vatu u džepu nošnje
Gojazni đaćić s biciklom drži hmelj i finu vatu u džepu nošnje.

Рис. 2. Скриншоты шрифтов без хинтинга и с ним

Из рекомендаций в области микротипографики как для веб-дизайнеров, так и для разработчиков можно отметить следующее:

- не следует использовать наклон в свойствах CSS (или так называемый «искусственный наклон»), так как он не имеет ничего общего с курсивным начертанием выбранного шрифта; если требуется курсивное начертание, то лучше подключить соответствующий шрифт с таким начертанием;
- не следует использовать подчеркивание в обычном тексте, так как в вебе таким образом чаще всего выделяется ссылка, что может ввести пользователя в заблуждение. При этом в

общей структуре текста ссылкам необходимо уделить особое внимание, они должны с легкостью выделяться из окружающего их контента;

- не следует устанавливать жирное начертание с помощью свойств CSS (или так называемый «искусственный жирный шрифт»); если требуется жирное начертание, то лучше подключить соответствующий шрифт с таким начертанием;
- чем меньше используемый шрифт, тем больше должны быть межбуквенные расстояния, чем крупнее – тем плотнее должны быть расположены буквы;
- не стоит использовать набор заглавными буквами больше чем на одну строку, такой текст сложно читать;
- следует выбирать шрифты с различимыми буквами, так как во многих шрифтах слишком легко перепутать аналогичные формы букв, в частности «i» и «L» (рис. 3), и маленькие расстояния между буквами, например, когда «t» и «n» выглядят как «m».



Рис. 3. Пример плохой различимости букв «i» и «L» в шрифте (а) и пример хорошей их различимости (б)

Литература

1. Reichenstein O. Web design is 95% typography [Электронный ресурс]. – Режим доступа: <https://ia.net/topics/the-web-is-all-about-typography-period/>, своб.
2. Latin M. Better web typography for a better web. – Penarth: Studio Two, 2017. – 215 p.
3. Dugonjić M. Designing For The Reading Experience [Электронный ресурс]. – Режим доступа: <https://www.smashingmagazine.com/2013/02/designing-reading-experience/>, своб.
4. Орлов А.С. Основные положения технологии проектирования пользовательских интерфейсов // Изв. ТРТУ: компьютерные технологии в инженерной и управленческой деятельности. – 2002. – № 5. – С. 34–38.
5. Bonneville D. The Big Book Of Font Combinations. – Rhode Island: BonFX Press, 2010. – 135 p.

**Лисакова Анастасия Владимировна**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: cocktail_rossini@mail.ru

УДК 004

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ ВОЗМОЖНОСТЕЙ ФРЕЙМВОРКОВ TWITTER
BOOTSTRAP И ZURB FOUNDATION****Лисакова А.В.****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 HTML/CSS фреймворки Twitter Bootstrap и Zurb Foundation. Проведено сравнение характеристик и возможностей изучаемых фреймворков.

Ключевые слова: HTML/CSS фреймворки, Twitter Bootstrap, Zurb Foundation, HTML.

В ранние годы большая часть стилей веб-страниц могла быть достигнута благодаря использованию HTML. В настоящее время как HTML5, так и CSS3 тесно взаимодействуют, чтобы добиться поразительных результатов в веб-дизайне, дизайне приложений и, в некоторых случаях, даже в разработке программного обеспечения. Тем не менее технологии верстки не стояли на месте, а разработчики стремились к упрощению и автоматизации процесса. Так появились HTML/CSS фреймворки.

Фреймворк представляет собой набор стандартизированных концепций, практик, критериев для решения проблем аналогичного характера. Он состоит из структурированной системы папок, состоящих из HTML, CSS, JS и других видов документов. Эти файлы могут использоваться для разработки веб-сайтов, особенно в качестве основы для его создания. Так как большинство веб-сайтов имеют сходную структуру, то целью фреймворков является предоставление общей структуры, которую не придется делать с нуля, а также код которой можно повторно использовать в других проектах. Таким образом, фреймворки позволяют разработчикам сократить большую часть работы и сэкономить достаточно времени.

Стандартные компоненты, входящие во фреймворк:

- исходный код CSS для создания сетки, что позволяет разработчику размещать различные элементы веб-сайта простым и универсальным способом;
- определения стилей типографии для элементов HTML;
- шаблоны HTML для форм, кнопок, оповещений, всплывающих окон и других многофункциональных компонентов;
- решения для кроссбраузерной верстки;
- создание типовых классов CSS, которые могут использоваться для стилизации дополнительных компонентов пользовательского интерфейса [1].

На данный момент существует широкий выбор HTML/CSS фреймворков.

Наиболее популярным является Bootstrap. Он содержит HTML- и CSS-шаблоны оформления для разметки, типографики, форм, кнопок, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения. Изначально он разрабатывался как внутренняя библиотека под названием Twitter Blueprint для социальной

сети Twitter. В 2011 году он был открыт для общего доступа под нынешним названием. Особой популярностью пользуется Bootstrap для создания одностраничников или «лендингов» (landing page) [2].

Zurb Foundation – одна из самых современных адаптивных платформ для создания интерфейсов в мире. Foundation содержит в себе все компоненты, необходимые для быстрого прототипирования, которое является основной целью использования CSS-фреймворка. Zurb – это название дизайнерской фирмы, которая и создала фреймворк.

Возможность быстрого создания сайтов выступает наивысшим приоритетом для фронтенд-разработчиков, ограниченных жесткими сроками сдачи проектов. Foundation был разработан для того, чтобы дать возможность быстро собирать прототипы и формировать код для современных сайтов и приложений, которые будут хорошо смотреться на любых устройствах. Foundation заявляется как полностью адаптивный фреймворк, нацеленный, прежде всего, на создание мобильных версий сайтов (Mobile First). Также фреймворк (Foundation for Emails) предлагает инструменты для адаптивной верстки электронных писем.

Bootstrap является безусловным лидером среди доступных фреймворков сегодня. Учитывая его огромную популярность, которая продолжает расти каждый день, можно быть уверенным, что этот инструмент будет в дальнейшем поддерживаться разработчиками. Bootstrap используется на Upwork, NPM, Microsoft.

Foundation второй большой и популярный фреймворк. Компания Zurb сопровождает его, обеспечивая хорошую поддержку и развитие этого фреймворка. Foundation используется на многих крупных сайтах, в том числе Facebook, Mozilla, Ebay, Yahoo!, и National Geographic. Также Foundation считается профессиональным фреймворком с поддержкой бизнес-компанией [3].

Если взять во внимание статистику Google Trends, то поисковой запрос «Bootstrap» намного активнее, чем «Foundation» (рисунок), несмотря на то что Bootstrap 4 вышел год назад [4].

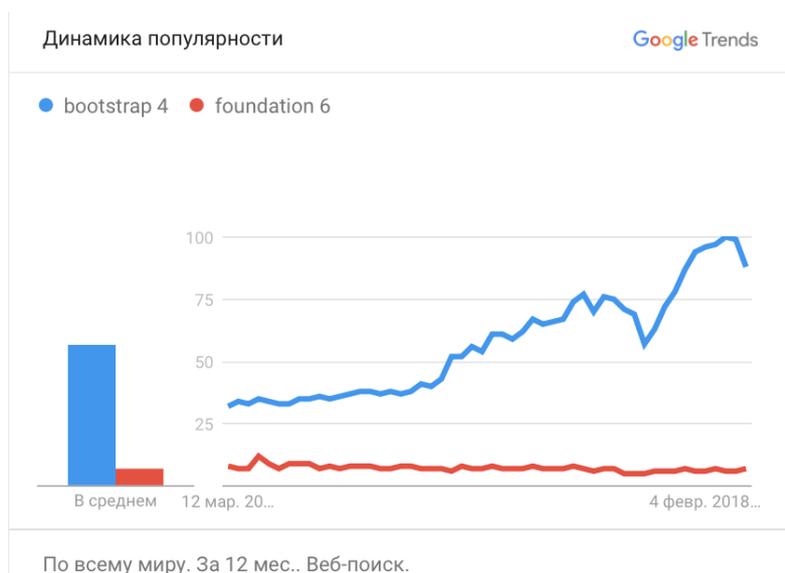


Рисунок. Динамика популярности поисковых запросов

Для проведения сравнительного анализа можно выделить следующие критерии: сообщество, используемый препроцессор, сетка, величина измерения, кроссбраузерность, сброс CSS.

1. Сообщество. Многие разработчики при выборе фреймворка или библиотеки анализируют их репозиторий на Github. Количество звезд на Github является одним из важнейших показателей популярности того или иного инструмента.

2. Препроцессор. Препроцессоры дают много возможностей в написании стилей для веб-страницы, упрощая и автоматизируя процесс. Выбор того или иного препроцессора обычно зависит от знаний разработчика.
3. Сетка. Сетка во фреймворках используется для размещения элементов на веб-сайте простым и удобным способом. Разные инструменты предлагают разные способы построения сетки.
4. Величина измерения. Основной величиной является px, но есть и удобные вариации такие, как rem, em и т.д. Их использование позволяет сделать веб-сайты более гибкими и адаптивными.
5. Кроссбраузерность. Важно, чтобы элементы на веб-сайте, построенные при помощи инструментов фреймворков, отображались одинаково во всех существующих (или используемых, популярных) браузерах.
6. Сброс CSS. Стили браузера по умолчанию могут мешать стилям, которые разработчик на самом деле применяет, поэтому был разработан сброс CSS, чтобы обеспечить совместимую базу для всех браузеров.

В таблице проведено сравнение Twitter Bootstrap и Zurb Foundation по указанным выше критериям.

Таблица. Сравнительный анализ Twitter Bootstrap и Zurb Foundation

Критерий	Twitter Bootstrap	Zurb Foundation
Сообщество	122522 звезды	27085 звезд
Препроцессор	В 3 версии использовался препроцессор Less, в 4 версии его изменили на Sass	Sass
Сетка	3 версия – Float. С 4 версии появилась Flexbox Grid	Float, Flexbox, XY Grid – любую из сеток можно подключить и использовать в проекте
Величина измерения	px – стандартная величина измерения	rem – величина измерения, которая зависит от font-size, установленного в html и body
Кроссбраузерность	Поддерживает последние версии всех основных браузеров и платформ. В Windows они поддерживают Internet Explorer 9-11 и Microsoft Edge. Не поддерживается IE-8 в 4 версии	Поддерживает последние версии всех основных браузеров и платформ. В Windows они поддерживают Internet Explorer 9-11. Но IE8 не поддерживается из-за некоторых свойств сетки, а также из-за JavaScript
Сброс CSS	reboot.css	normalize.css

Как видно из таблицы, Twitter Bootstrap обходит Foundation по популярности сообщества (122522 звезды против 27085). В Foundation используется более удобная величина измерения – rem. Она зависит от font-size на всем веб-сайте, что позволяет масштабировать элементы путем уменьшения только font-size. Bootstrap в своих версиях предлагает использование одного вида сетки, когда в Foundation можно выбрать на свое усмотрение. Оба фреймворка поддерживают современные браузеры, а также Internet Explorer 9-11. Для сброса стилей Bootstrap использует reboot.css, который основан на normalize.css, используемый в Foundation.

Таким образом, выбор Bootstrap или Foundation зависит от разработчика, используемых им технологий и от самого проекта, над которым ведется работа. Тем не менее, Bootstrap все еще является самым популярным HTML/CSS фреймворком, особенно в России.

Литература

1. Keycdn.com. Bootstrap vs Foundation – Top 2 CSS Frameworks [Электронный ресурс]. – Режим доступа: <https://www.keycdn.com/blog/bootstrap-vs-foundation/>, своб.
2. Bootstrap. About [Электронный ресурс]. – Режим доступа: <https://getbootstrap.com/docs/4.0/about/overview/>, своб.
3. ZURB Foundation. Foundation for Sites 6 [Электронный ресурс]. – Режим доступа: <https://foundation.zurb.com/sites.html>, своб.
4. Google Trends. Bootstrap vs Foundation [Электронный ресурс]. – Режим доступа: <https://trends.google.ru/trends/explore?q=bootstrap%204,foundation%206>, своб.

**Литвинов Кирилл Владимирович**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4108Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: SquertelSquertel@mail.ru

**Сокуренок Юрий Андреевич**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», к.т.н., доцент

e-mail: kpd@limtu.ru

**Государев Илья Борисович**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: goss@corp.ifmo.ru

УДК 004.7**ОБЗОР ТЕОРЕТИЧЕСКИХ И ЭМПИРИЧЕСКИХ ОСНОВ ВЕБ-РАЗРАБОТКИ
ДЛЯ ИНТЕРНЕТА ВЕЩЕЙ****Литвинов К.В., Сокуренок Ю.А., Государев И.Б.****Научный руководитель – к.т.н., доцент Сокуренок Ю.А.**

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

В работе рассмотрены теоретические и эмпирические основы веб-разработки для Интернета вещей. Работа содержит исследование материалов, учебников, научных журналов, статей и справочных данных сети Интернет. Представлены перспективы развития сферы Интернета вещей, а также выделены основные проблемы информационной безопасности и интерфейсов данной сферы.

Ключевые слова: Интернет вещей, анализ, веб-интерфейс, прогнозирование, IoT, информационная безопасность.

Первое упоминание термина Интернет-вещей (Internet of Things, IoT) было в 1999 году, он был разработан Кевином Аштоном [1], работавшим в то время в компании «Procter&Gambel». Зарождалась идея о том, что в повседневные вещи, такие как: холодильник, автомобиль, одежда – можно встроить беспроводные датчики, связанные друг с другом.

Сейчас наблюдается рост компьютерных коммуникаций, так как потребитель активно использует Wi-Fi, 3G, 4G для обмена различного рода информацией. Интернет вещей активно развивается при помощи нескольких факторов:

– развитие облачных технологий и мобильных устройств;

- способность анализировать большие объемы данных и превращать их в удобную для пользователя информацию;
- способность комбинировать технологии (программное обеспечение и аппаратную часть).
Сочетание всех вышеперечисленных факторов позволяет достичь всеобщей «подключенности» (smartconnectivity).

Основная мысль Интернета вещей заключается в организации и взаимодействии разнообразных устройств в окружающей нас среде, передачи информации, генерируемой этими устройствами и предоставление бесперебойного соединения.

Модель IoT предполагает, что у каждого устройства должно быть в наличии, как минимум, устройства для коммуникации и других возможностей. Например, возможность ввода, хранения, обработки данных.

Компания Gartner прогнозирует [2], что в 2017 году по всему миру будет использоваться 8,4 млрд подключенных устройств к Интернету вещей – это на 31% больше, чем было в 2016 году. Общие расходы и услуги рынка на различные устройства в 2017 году достигнут приблизительно \$2 трлн.

Основные регионы, которые обеспечивают развитие Интернета вещей, являются: Китай, Северная Америка и Западная Европа. На их долю вкуче будет приходиться 67% всей установленной базы IoT в 2017 году.

В 2017 году расходы различных компаний на IoT-оснащение составили \$694 млрд, а потребительские расходы – \$725 млрд.

Компания Forrester утверждает [2], что IoT-технологии в общем разнообразны и незрелы. Стандарты на данный момент находятся на стадии зарождения, так как продавцы всего лишь пару лет занимаются этим вопросом. Технологии безопасности IoT все еще в стадии создания без применения в конкретных продуктах.

В отчете для TeacheRadar, компания Forrester упомянула, что 23% компаний уже используют IoT, а 29% планируют использовать в течение следующего года.

Международный экономический форум (World Economic Forum, WEF) [3] представил список, чего стоит ожидать от Интернета вещей в 2018 году:

- мировые расходы Интернета вещей в 2017 году достигли \$674 млрд, а в 2018 году увеличатся еще на 14,6% до 772,5 млрд;
- азиатско-Тихоокеанский регион (за исключением Японии) будет географическим регионом с наибольшими расходами на IoT в 2018 году – \$312 млрд, за которым следует Северная Америка – 171 млрд. Китай станет страной с наибольшим объемом расходов на IoT в 2018 году;
- потребительские расходы на IoT в 2018 году достигнут \$62 млрд, что станет четвертым по величине сегментом отрасли. Ведущие потребительские расходы будут связаны с «умным домом», в том числе домашней автоматикой и безопасностью умных устройств.

Из представленных прогнозов можно сделать вывод, что Интернет вещей стремительно развивается, но также можно выделить основные проблемы: вопрос безопасности, многообразие различных протоколов, отсутствие общепринятых стандартов и потока данных, которые современные сети просто не выдерживают.

В первую очередь IoT – это определенный набор устройств, которые должны уметь общаться друг с другом (передавать данные другим устройствам), где-то хранить эти данные и каким-то образом обрабатывать их.

Информационная безопасность Интернета вещей. В научной статье А.Ю. Лузянина и Н.Ш. Аманова [4] рассматривается проблема информационной безопасности (ИБ) в Интернете вещей. Авторы статьи уделяют особое внимание приему и передаче пользовательских данных в открытых каналах связи. Поднимается проблема отсутствия интереса со стороны разработчиков к защите пользовательских данных. Проблема безопасности настолько серьезна, что рассматривается на самом высоком уровне. В отчете

всемирного экономического форума говорится, что выработка единого подхода к решению проблемы безопасности – самый необходимый шаг для развития Интернета вещей. Также в работе М.Н. Соколова, К.А. Смолянинова и Н.А. Якушева [1] рассматривается проблема ИБ в сфере Интернета вещей. Они утверждают, что данная проблема особенно актуальна из-за нескольких причин:

- прогнозируется рост потребности пользователей в IoT;
- широкое распространение беспроводных технологий и межмашинного обмена;
- развитие технологий облачных вычислений;
- начало перехода на IPv6.

Интерфейсы Интернета вещей. В своей работе О.Ю. Никифоров [5] рассматривает вопрос актуальности и понимания новых интерфейсов для Интернета вещей. Рассматриваются такие виды интерфейсов, как:

- интерфейс «машина-машина» (M2M). Данный интерфейс является физической основой для Интернета вещей. Он служит для взаимодействия вещей с вещами или машин с машинами;
- интерфейс «машина-инфраструктура» (M2I) служит для взаимодействия вещей с инфраструктурой внешней среды. Примером реализации данного интерфейса могут послужить интеллектуальные транспортные сети;
- интерфейс «машина-природа» (M2N). Этот интерфейс служит для взаимодействия машин с природой. Примером активного использования данного интерфейса может быть «умный дом»;
- для взаимодействия человека с самим собой. Интерфейс «человек-я» (P2H) является следствием развития сети Интернет, который благодаря Интернету вещей получил многократное усиление: умные часы сигнализируют о связанной активности в социальных сетях, мобильные гаджеты позволяют быть онлайн все время;
- интерфейс «человек-тело» (P2B). Данный интерфейс служит для взаимодействия человека с собственным телом. Этот интерфейс существовал давно, но Интернет вещей перевел его на новый более качественный уровень, который позволяет реализовать такие функции, как удаленный мониторинг и контроль здоровья;
- для взаимодействия человека с вещами. Интерфейс «человек-машина» (P2M). Данный интерфейс существует на протяжении всей истории развития человечества, но Интернет вещей позволяет взглянуть на это взаимодействие с другой стороны. Допустим есть ручка. Ручка – это ручка, но если на ней размещен штрих-код, на котором размещена определенная информация, считывая ее с помощью смартфона, можно коммуницировать с ручкой.
- интерфейс человек-человек» (P2P). Этот интерфейс служит для взаимодействия человека с человеком. Мобильные гаджеты делают процесс установки удаленного соединения между людьми мгновенно, а новые нейроинтерфейсы позволяют реализовать концепцию «коллективного разума».

В работах [6, 7] авторы делают акцент на том, что интерфейсов на сегодняшний день большое количество. Эта влечет за собой проблему масштабируемости Интернета вещей, поскольку на данный момент нет определенного стандарта по разработке интерфейсов для IoT, чтобы вещи могли спокойно обмениваться данными друг с другом. Также в данных работах уделяется большое внимание психологическому аспекту разработчиков. Так как почти никто не хочет предоставлять свои решения в открытый доступ.

Adrian Bridgwater подчеркивает, что отсутствие API-стандартов приводит к проблемам соединения, безопасности, обновлению данных, а также ограничениям производительности. Много времени будет потрачено на попытки подключить устройства к сети и заставить их взаимодействовать друг с другом без API-стандартов.

В результате можно сделать вывод, что на текущий момент нет единого протокола

передачи данных. Это влечет за собой отсутствие каких-либо мер по защите персональных данных пользователя. Производители оборудования не заинтересованы в развитии и защите персональных данных. Можно отметить неготовность текущих средств защиты работать в реальном времени. Также можно выделить проблемы стандартизации API для Интернета вещей. Без этих стандартов разработка приложений, обмен данными и взаимодействие между различными интерфейсами будет затруднено в разы.

Литература

1. Ермакова Н.Е., Константинов Е.С. Интернет вещей // Сб. научных трудов вузов России «Проблемы экономики, финансов и управления производством». – 2015. – № 37. – С. 129–132.
2. Проблемы и перспективы Интернета вещей [Электронный ресурс]. – Режим доступа: <https://rb.ru/opinion/russian-iot/> (дата обращения: 09.11.2017).
3. The Global Competitiveness Report 2016–2017 // World economic forum [Электронный ресурс]. – Режим доступа: http://www3.weforum.org/docs/GCR2016-2017/05FullReport/TheGlobalCompetitivenessReport2016-2017_FINAL.pdf (дата обращения: 31.11.2017).
4. Лузянин А.Ю., Аманов Н.Ш. Информационная безопасность интернета вещей // Инновационная наука. – 2016. – № 7-8. – С. 62–63.
5. Никифоров О.Ю. Интерфейсы Интернета вещей // Научный альманах. – 2015. – № 9(11). – С. 1344–1347.
6. Соколов М.Н., Смолянинова К.А., Якушева Н.А. Проблемы безопасности Интернет вещей: обзор // Вопросы кибербезопасности. – 2015. – № 5(13). – С. 32–35.
7. Елизаров М.А. Перспективы и проблемы развития рынка интернета вещей [Электронный ресурс]. – Режим доступа: https://revolution.allbest.ru/programming/00862173_0.html, своб.

**Лобач Ирина Николаевна**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4106Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: Irka060595@gmail.com

**Шуклин Дмитрий Анатольевич**

Год рождения: 1979

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: do@limtu.ru

УДК 004.928

**ИСПОЛЬЗОВАНИЕ АНИМАЦИОННЫХ ЭФФЕКТОВ В ДИЗАЙНЕ
ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ВЕБ-САЙТОВ****Лобач И.Н., Шуклин Д.А.****Научный руководитель – к.педагог.н., доцент Шуклин Д.А.**

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

В работе рассмотрены виды использования анимационных эффектов в дизайне пользовательских интерфейсов веб-сайтов. Даны ключевые определения понятий «Анимация», «Интерактивная анимация». Дано краткое представление о технологиях использования анимации в веб-дизайне.

Ключевые слова: анимация, интерактивная анимация, веб-дизайн, пользовательский интерфейс, технологии, веб-сайт.

Анимация в веб-дизайне на сегодняшний день является крайне важным и полезным приемом. Сейчас разработчики интерфейсов довольно часто используют анимационные эффекты для улучшения навигации на сайте и для повышения общего уровня удобства пользования веб-сайтом. Также анимация используется для привлечения внимания пользователя к важным деталям. Анимационные эффекты используют, чтобы подчеркнуть кликабельность элемента на фоне остальных вещей.

Взаимодействие пользователей с современными веб-сайтами серьезно завязано именно на анимации. Она способна сообщать о неких состояниях страницы, направлять внимание. И помогает пользователю увидеть результат его действий, может влиять на его поведение.

Классическое определение анимации изложено зарубежными авторами Р. Бекером и И. Смоллом [1]. Анимация – это динамическое визуальное изложение, форма и структура, развивающиеся с течением времени.

Существует также отдельный класс анимации – интерактивная анимация. Определение интерактивной анимации изложено в статье С.В. Сидорова [2]. Интерактивная анимация – это анимация, реагирующая на действия пользователя. В отличие от обычной анимации, которую мы смотрим, являясь зрителями, с интерактивной анимацией необходимо взаимодействовать, выстраивая это взаимодействие последовательно.

Историю использования анимации в вебе связывают с технологиями. Ранее анимация осуществлялась в основном при помощи платформы Flash, не интерактивных изображений в формате GIF и скриптов на языке программирования JavaScript. Долгое время основной задачей анимации в веб-интерфейсах было привлечение внимания пользователей. Вследствие этого анимация воспринималась скорее как развлекательный прием.

Сегодня анимацию создают с помощью более легковесного JavaScript и CSS-кодирования, которое помогает добавлять движущиеся элементы на сайт, при этом не перегружая его. И, что немаловажно, анимацию теперь используют для улучшения юзабилити сайта, а не просто для развлечения. Она является отличным инструментом для веб-дизайнеров, помогая делать сайты лучше и проще в использовании, и является основой эффективного интерактивного дизайна.

В дизайне пользовательских интерфейсов веб-сайтов анимацию можно применять для различных целей и задач, стоит рассмотреть несколько основных видов анимации, используемых в веб-дизайне, такие как:

1. анимация при загрузке данных. Самая распространенная ситуация использования анимации – это отвлечение пользователя во время загрузки контента. Все дело в том, что анимация в такие моменты воздействует на то, как пользователи воспринимают время. Она создает ощущение того, что действие выполняется быстрее, чем на самом деле;
2. анимация процессов и пошаговых операций. Такой тип анимации показывает выполнение некой линейной последовательности действий, не зависящей от пользователя. Самый распространенный пример – это классическая полоса загрузки. Также анимацию используют для процессов, предусматривающих выполнение пользователем некоторых последовательных действий;
3. анимация и каркасное отображение веб-страниц. При каркасном отображении веб-страниц необходимо наличие пространства, которое заполняется загружаемой информацией постепенно. При плавном наполнении страницы, выполнение действий ощущается быстрее. Эту технику можно комбинировать совместно с ненавязчивой анимацией процесса загрузки, она позволяет удерживать внимание пользователя;
4. анимированная реакция на действия пользователя. Дизайн взаимодействий с пользователем предусматривает обратную связь, она сообщает о результатах взаимодействия. Анимация при наведении курсора на интерактивный элемент – это один из самых распространенных примеров визуальной обратной связи. Когда пользователь не уверен в назначении элемента управления, он наводит на него курсор мыши. В этот момент анимация элемента, на интуитивном уровне, сообщает пользователю о том, что с объектом можно взаимодействовать. Это справедливо для сайтов, просматриваемых на персональных компьютерах. Для мобильных версий сайтов, кнопки и другие интерактивные элементы оснащаются визуальными подсказками. Подсказки указывают на то, что с элементом можно взаимодействовать еще до того, как пользователь коснется его. А после касания дают визуальную обратную связь;
5. привлечение внимания с помощью анимации. Известно, что наш взгляд привлекает движение. Данным фактом можно пользоваться для того, чтобы направлять внимание пользователя к определенным действиям. Например, при вводе данных в форму, с помощью анимации можно сообщить пользователю о правильно и неправильно заполненных полях. Такой тип анимации позволяет пользователю моментально понять то, к каким результатам привело его действие;
6. переходы и взаимоотношения состояний. При помощи такого типа анимации можно обеспечивать связь состояний приложения и предотвращать слишком быстрые переходы между экранами. Правильно спроектированные переходы позволяют пользователям понять, на чем следует сосредоточиться в конкретный момент. Также анимацию можно использовать для того, чтобы сделать более очевидные переходы между экранами

приложения. В итоге пользователь будет точно понимать, где он был в начале действия, и где оказался после его завершения;

7. плавное изменение состояний. Для указания смены состояний страницы необходима анимация переходов. Данный тип анимации помогает пользователю быть в курсе того, где именно он находится, при щелчке по ссылке, которая ведет к другому разделу той же самой страницы. Пользователю тяжело воспринимать внезапное изменение состояния страницы, зачастую оно выглядит ненатурально. Например, в механизме навигации по сайту можно использовать ненавязчивую анимацию, которая будет помогать пользователю ориентироваться в статье или новостной ленте, он будет видеть, где расположен просматриваемый раздел страницы [3].

Дизайн – это больше чем визуальное представление данных, это управление взаимодействием пользователя и веб-страницы. Таким образом, анимация играет чрезвычайно важную роль для передачи информации.

Литература

1. Backer R., Small I. Animation at the Interface [Электронный ресурс]. – Режим доступа: <http://ron.taglab.ca/papers/B3.pdf>, своб.
2. Сидоров С.В. Возможности интерактивной анимации в учебном процессе [Электронный ресурс]. – Режим доступа: http://si-sv.com/publ/1/vozmozhnosti_interaktivnoj/14-1-0-524, своб.
3. Babich N. Popular Web Animation Techniques [Электронный ресурс]. – Режим доступа: <http://babich.biz/popular-web-animation-techniques/>, своб.



Ломтев Илья Анатольевич

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: ilylomte7@gmail.com

УДК 004

**ИССЛЕДОВАНИЕ ПРОТОКОЛОВ ПЕРЕДАЧИ ДАННЫХ ДЛЯ СИСТЕМ
МОНИТОРИНГА**

Ломтев И.А.

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 протоколы передачи данных, технологии: WebSocket, AJAX, Comet, HTML SSE, их возможности, области применения сравниваются между собой относительно применения в системах мониторинга.

Ключевые слова: системы мониторинга, протоколы передачи данных, WebSocket, AJAX, Comet, HTML SSE.

Исторически сложилось так, что создание веб-приложений, которым требуется двунаправленная связь между клиентом и сервером (например, мгновенный обмен сообщениями и игровые приложения), требовало злоупотребления HTTP-запросами, в виде отдельных HTTP-вызовов [1], для опроса сервера и обновления данных, при отправке уведомлений.

Это приводит ко множеству проблем [1]:

- сервер вынужден использовать несколько разных TCP-соединений для каждого клиента: один для отправки информации клиенту и новый для каждого входящего сообщения;
- проводной протокол имеет высокие накладные расходы, причем каждое сообщение «клиент-сервер» имеет HTTP-заголовок;
- клиентский скрипт вынужден поддерживать сопоставление исходящих подключений к входящему соединению для отслеживания ответов.

Более простым и верным решением было бы использовать одно TCP-соединение для трафика в обоих направлениях. Это обеспечивает протокол WebSocket. В сочетании с API WebSocket он предоставляет альтернативу HTTP-запросам для двусторонней связи веб-страницы с удаленным сервером.

Этот метод может использоваться для различных веб-приложений:

- игры;
- биржевые приложения;
- многопользовательские приложения с одновременным редактированием;
- пользовательский интерфейс, предоставляющий серверные службы в режиме реального времени и т.д.

Протокол WebSocket предназначен для замены существующих двунаправленных коммуникационных технологий, которые используют HTTP в качестве транспортного уровня для использования существующей инфраструктуры (прокси, фильтрация, аутентификация). Такие технологии были реализованы как компромиссы между эффективностью и надежностью, поскольку изначально HTTP не предназначался для

двунаправленной связи. Протокол WebSocket решает задачи существующих двунаправленных HTTP-технологий в контексте существующей инфраструктуры HTTP; как таковой, он предназначен для работы через HTTP-порты 80 и 443, а также для поддержки HTTP-прокси и других промежуточных технологий, даже если это подразумевает определенную сложность, характерную для текущей среды. Однако дизайн не ограничивает использование WebSocket для HTTP, а в будущих реализациях может использоваться упрощенный handshake по выделенному порту без повторного использования всего протокола. Этот последний момент важен, поскольку шаблоны трафика интерактивных сообщений не соответствуют стандарту HTTP-трафика и могут вызвать необычные нагрузки на некоторые компоненты.

Перед тем как начать обмен данными, клиент и сервер согласуют параметры соединения, а именно: способ шифрования, версию используемого протокола, могут проверить сертификаты, если это нужно. Такая схема начала соединения называется Handshake [2].

Протокол состоит из двух частей: handshake (рисунок) и передачи данных.

```
1 // Запрос от браузер
2 GET /chat HTTP/1.1
3 Host: server.example.com
4 Upgrade: websocket
5 Connection: Upgrade
6 Origin: http://javascript.ru
7 Sec-WebSocket-Key: Iv8io/9s+lYFgZwcXczP8Q==
8 Sec-WebSocket-Version: 13
9
10 // Ответ сервера
11 HTTP/1.1 101 Switching Protocols
12 Upgrade: websocket
13 Connection: Upgrade
14 Sec-WebSocket-Accept: hsB1buDTkk24srzE0TBULZAlC2g=
```

Рисунок. Handshake от клиента и от сервера

Ведущая строка от клиента соответствует формату Request-Line. Ведущая строка с сервера соответствует формату Status-Line. Операции «Запрос-строка» и «Строка-строка» определены в RFC2616.

Неупорядоченный набор полей заголовка приходит после первой строки в обоих случаях.

Как только клиент и сервер отправили свои handshakes, и если handshakes были успешными, начинается передача данных. Это двухсторонний канал связи, где каждая сторона может независимо от другого отправлять данные по своему усмотрению.

После успешного handshake клиенты и серверы передают данные в обоих направлениях, как «сообщения». Сообщение состоит из одного или нескольких кадров. Сообщение WebSocket не обязательно соответствует кадрированию определенного сетевого уровня, поскольку фрагментированное сообщение может быть объединено или разделено промежуточной технологией.

Кадр имеет связанный тип. Каждый кадр, принадлежащий тому же сообщению, содержит один и тот же тип данных. Существуют типы текстовых данных (которые интерпретируются как текст UTF-8), двоичные данные (интерпретация которых оставлена до приложения) и управляющие кадры (которые не предназначены для переноса данных для приложения, но вместо этого для сигнализации на уровне протокола, например, чтобы сигнализировать о том, что соединение должно быть закрыто).

Мониторинг подразумевает трансляцию считываемых показателей непрерывно и с наименьшей задержкой. Для того чтобы это реализовать, нужно обратиться к

соответствующим технологиям. В настоящее время подходят следующие технологии: AJAX, стек технологий Comet, HTML5 SSE и WebSocket. Использование AJAX снизит нагрузку на сервер путем уменьшения объема данных в запросе, а значит, будет происходить лишь частичное обновление данных на клиенте. Принцип работы приложений на AJAX – «запрос-ответ»: на сервер приходит запрос каких-либо новых данных, после того, как сервер его обрабатывает, он формирует ответ и отправляет его клиенту. Клиент, получив ответ, преобразует его и мутирует пользовательский интерфейс. Но у AJAX есть важный минус, он не может получить обновления от сервера без запроса. Таким образом, нужно постоянно отправлять запросы и ждать ответ. Интервал, соответственно, должен быть небольшим, чтобы не упустить важное обновление информации. Но вот заранее неизвестно, придут ли новые данные, поэтому большое количество ответов на запросы будет пустыми. Однако нагрузку оказывает любой запрос. Следовательно, эти запросы будут оказывать лишнюю нагрузку на оборудование.

AJAX является довольно давней разработкой, он широко известен. К примерам использования AJAX можно отнести: автоматическое сохранение текстов в веб-редакторах через определенный промежуток времени или после каждого нажатия, обновление входящих писем, чаты [3]. Использование AJAX приносит определенные риски и сложности разработки защищенных систем, так как есть ряд уязвимостей, таких как, например, отправка вредоносного JavaScript-кода, подмена сессии и др.

Comet – модель веб-приложения, в котором постоянное HTTP-соединение позволяет получать данные от сервера без запроса со стороны клиента [4]. Веб-приложения, которые реализуют эту модель называются Comet-приложениями. В Comet-приложениях клиент умеет взаимодействовать с сервером через постоянное либо long polling HTTP-соединение. Реализации Comet делятся на потоковые, при них поддерживается постоянное соединение, и long polling, длительное соединение. Во-втором случае ожидается ответ в течение лимита времени по открытому соединению, по истечению времени соединение переустанавливается. Long polling позволяет после отправки HTTP-запроса к серверу ждать данных в ответе. Соединение поддерживается пока с сервера не будут отправлены данные. После получения ответа соединение разрывается и устанавливается заново. Иногда используется буферизация – данные накапливаются для передачи большого объема информации за одно подключение. После получения данных от сервера, открывается новое соединение. Обычно длительность такого соединения не больше 5 мин, сервера сами обрывают слишком долгие соединения, потому что HTTP изначально не приспособлен для такого использования. Использование long polling благоразумно в системах, где изменения, новые события, происходят не часто, например, обновление новостных лент.

HTML5 SSE (Server Sent Events) устроен иначе: со стороны клиента уходит запрос на сервера по HTTP. В ответ приходит страница с JavaScript-кодом, который устанавливает соединение с сервером. Затем сервер посылает сообщения, когда у него появляются новые данные. В итоге, один запрос от клиента и много сообщений от стороны сервера. Канал связи HTML5 SSE односторонний.

Потоковая реализация (streaming) у модели Comet устроена так: в скрытый контейнер, iframe, постоянно приходят данные, чтобы поддерживать соединение открытым. В какой-то момент от сервера приходит JavaScript-код, который сразу выполняется. При данном подходе соединение будет оставаться открытым и не будет прерываться после каждого сообщения. Таким образом, новые сообщения приходят без задержек. Применение потоковой реализации есть, например, на биржевых ресурсах, где обновление котировок происходит часто, а из-за этого long polling не подходит. С такой реализацией не получится реализовать обработку ошибок. Также не получится различать разрыв соединения и отсутствие сообщений с сервера.

Ну и наконец, протокол полнодуплексной связи WebSocket использует концепцию «сообщение-сообщение», она отличается от «запрос-ответ» тем, что клиент и сервер –

равноправные участники обмена информацией. Протокол предполагает отправку сообщений от клиента серверу и наоборот, не дожидаясь обработки и ответа. Протокол открывает постоянный канал двусторонней связи, это позволяет получать все обновления клиенту от сервера практически мгновенно, как только данные появились на сервере, в отличие от AJAX. Использование WebSocket позволяет, как и с SSE и с Comet, уменьшить задержку в получении новых данных. А также, в отличие от них, протокол реализует не просто обновление данных от сервера без запроса, но и возможность посылать сообщения с новыми данными от клиента серверу.

Также можно устанавливать защищенное соединение. В этом случае соединение открывается поверх HTTPS, а не HTTP. Применяется в системах, которые предполагают интенсивный обмен данными: CRM-системы, многопользовательские сетевые игры, социальные сети, системы биржевой торговли и др. Кроме того, при использовании WebSocket участникам соединения не нужно обрабатывать и передавать большие заголовки и метаданные, как это происходит при использовании AJAX, что, в свою очередь, ускоряет работу системы. Протокол лишен уязвимостей, которые есть у AJAX.

Применение протокола WebSocket позволит улучшить качество работы, скорость реагирования, качество труда. Это технология предназначена для подобных задач. Ее применение, вместо вынужденных решений, поможет улучшить качество работы [5].

Литература

1. RFC 6455 – The WebSocket Protocol (RFC6455) [Электронный ресурс]. – Режим доступа: <http://www.faqs.org/rfcs/rfc6455.html>, своб.
2. WebSocket. MDN [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/ru/docs/Web/API/WebSocket>, своб.
3. Медведев Ю.С., Терехов В.В. К вопросу о достижении максимальной производительности AJAX-приложений // Вестник Адыгейского государственного университета. – 2013. – № 4(125). – С. 121–124.
4. Тюлькин М.В., Капгер И.В., Кротова У.Л., Кротов Л.Н. Разработка архитектуры и организация информационных потоков в comet-серверах для web-приложений модели comet со схемой взаимодействия websocket. Описание comet-сервера // Вестник ИЖГТУ им. М.Т. Калашникова. – 2016. – № 2. – С. 150–153.
5. Правильный мониторинг. Хайлоад [Электронный ресурс]. – Режим доступа: <https://ruhighload.com/post/%D0%9F%D1%80%D0%B0%D0%B2%D0%B8%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9+%D0%BC%D0%BE%D0%BD%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3+>, своб.



Лунев Алексей Витальевич

Год рождения: 1993

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: lunev1905@gmail.com

УДК 004.92

**АЛГОРИТМИЧЕСКИЙ ДИЗАЙН, ЕГО ОСОБЕННОСТИ И НЕДОСТАТКИ,
ИНСТРУМЕНТЫ РЕАЛИЗАЦИИ**

Лунев А.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 проведен анализ систем алгоритмического дизайна и их возможностей, выявлены преимущества и недостатки их применения в разработке web-приложений. Выделены несколько важных проблем использования инструментов алгоритмического дизайна и проектирования.

Ключевые слова: алгоритмический дизайн, web-дизайн, проектирование, интерфейс, web-сайт, web-приложения.

Актуальность работы обусловлена тем, что с развитием информационных технологий, таких как алгоритмы, нейронные сети, искусственной интеллект, появляется больше возможностей и инструментов для дизайна и проектирования web-приложений. Инструменты, которые помогают упростить построение интерфейса, подготовку графики и контента, а также персонализацию продукта, становятся автономнее, и участие человека сводится к минимуму. В последнее время такие инструменты стали доступнее, таким образом, появилась возможность использовать системы алгоритмического дизайна в повседневной работе.

Главное отличие от типичных инструментов дизайна и проектирования состоит в том, что алгоритмические системы полуавтономно формируют и выбирают варианты решений, что меняет способ взаимодействия пользователя с системой. Программа воспринимается не как средство, а как полноправный участник процесса [1].

В настоящее время алгоритмические модели базируются на значительном теоретическом ядре и практическом опыте. Часто применяются такие методы, как фракталы, «математический хаос», нейросети, рандомизация, клеточный автомат, системы Линденмайера (L-системы), «искусственная жизнь» и другие.

Можно отметить ряд основных шагов взаимодействия пользователя с системой алгоритмического дизайна:

1. формулирование задачи – представление результата, который пользователь рассчитывает получить в итоге;
2. установка характеристик – в систему устанавливаются параметры, которым должны отвечать генерируемые решения;
3. генерация – программа, делая упор на заданные параметры и установленные в нее алгоритмы, перебирает сочетания и визуализирует процессы и объекты. Алгоритмы обеспечивают «осмысленность» генерируемых объектов;
4. отбор объектов – пользователь дает оценку сформированным вариантам и подбирает подходящий ему вариант. Оценка строится на таких особенностях человека как вкус и

визуальное восприятие – а они слабо формализуемы, что приводит к тому, что алгоритмические системы не смогут заменить специалистов.

Примеры инструментов алгоритмического дизайна представлены в таблице.

Таблица. Инструменты алгоритмического дизайна и их возможности

Инструмент	Описание возможностей
NodeBox	Оффлайн-программа для MacOS, использует алгоритмические решения для формирования интерфейсов, спрайтов и графики, включающие постоянно изменяющиеся данные. Система дает возможность дизайнеру устанавливать характеристики генерации на основе блок-схемы и быстро получать результат при их изменении.
Rene	Проект дизайнера и инженера из Airbnb Джона Голда дает возможность оценивать и сравнивать различные виды комбинаций и кеглей, установив в системе некоторые ограничения.
Prototipo	Система-генератор шрифтов, которая успешно ввела алгоритмические технологии. Система похожа на Rene.
uKit AI	Российский онлайн-сервис, приспособливает страницы сайта к современным визуальным и техническим требованиям веба. На выходе получается новая, адаптивная версия front-end, сгенерированная на базе материалов с предыдущей версии сайта. Имеется возможность прикрепления к существующему домену.
pix2code	Датский стартап Uizard Technologies и нейросеть pix2code, способная определять макет, скриншот либо изображение интерфейса и формировать готовую интерактивную страницу с кодом, оформлением и графическими элементами, что позволяет автоматизировать однообразную верстку.
CMS Chorus	VoxMedia, в собственной CMS, реализовала механизм формирования главной страницы по следующему принципу – из подборки паттернов представления контента система выбирает гармоничные варианты, после дает оценку их возможной результативности и подбирает наилучший. Данные действия эффективнее и быстрее, чем ручная работа редактора.
CMS TheGrid	Самостоятельно подбирает шаблоны, обрабатывает фотографии, оформляет контент. Проводит А/В-исследования для поиска оптимального решения. Продукт уже пару лет в закрытой бете, и судить о нем можно было только по рекламным материалам и статьям.
WixAdvanced Design Intelligence	Полуавтоматический метод формирования сайтов, имеющий схожесть с TheGrid. Обучение алгоритма основывается на множестве образцов современных сайтов. Помимо этого, чтобы лучше попадать в стиль, система принимает во внимание тематику разрабатываемого сайта.

Системы Wix и TheGrid выступают в роли дизайнера-эксперта. Но абсолютный отказ от дизайнера приведет к однообразным и не всегда удовлетворяющим результатам (но в любом случае поднимет общий уровень качества). Если рассматривать эту работу как конгломерацию дизайнера с компьютером, то можно избавиться от части рутинной работы.

Дизайнер может помочь команде воплотить начальный макет и идею в целостный интерфейс с понятной логикой работы, информационной архитектурой и визуальным стилем. По ходу работы принимается огромное количество решений, выполняется решение поставленных задач, причем многие из них невозможно описать четкими алгоритмами. Поступающие условия могут быть ошибочны или противоречивы в деталях, дизайнер помогает урегулировать данные проблемы [2].

Оптимальным вариантом на текущий момент является совместная работа дизайнера и системы. Дизайнер и разработчик обрисовывают логику обработки поступающих сигналов: контента, контекста, информации о пользователе и его действиях, а дальше система создает интерфейс на базе готовых паттернов и принципов. Все это дает возможность достичь тонкой подстройки под определенную ситуацию и без потребности ручным способом прорисовывать и создавать множество состояний экрана. Быстро адаптировать интерфейс под разные платформы и устройства, пусть даже в примитивном виде [3–5].

Литература

1. Ветров Ю. Об интерфейсах [Электронный ресурс]. – Режим доступа: <http://www.jvetrau.com/algorithmic-design/> (дата обращения: 15.11.2017).
2. Я – дизайнер [Электронный ресурс]. – Режим доступа: <http://www.iamdesigner.pro/2015/11/dizajn-interfejsov-umer.-da-zdravstvuet-produktovuj-dizajn.html> (дата обращения: 15.11.2017).
3. Taking the robots to design school [Электронный ресурс]. – Режим доступа: <http://www.jon.gold/2016/05/robot-design-school/> (дата обращения: 15.11.2017).
4. Кирсанов Д. Веб-дизайн. – СПб.: Символ-Плюс, 2007. – 368 с.
5. Гэллоуэй А.Р. Неработающий интерфейс // Медиа: между магией и технологией. – 2014. – С. 252–288.

**Магнитова Мария Владимировна**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: magnitova@gmail.com

**Шуклин Дмитрий Анатольевич**

Год рождения: 1979

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: do@limtu.ru

УДК 004**АНАЛИЗ И КЛАССИФИКАЦИЯ КРИТЕРИЕВ ОЦЕНКИ КОММЕРЧЕСКИХ
ВЕБ-РЕСУРСОВ****Магнитова М.В., Шуклин Д.А.****Научный руководитель – к.ф.-м.н. Соловьев И.А.**

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

В работе выявлены требования, предъявляемые к современному коммерческому веб-ресурсу, на основании которых были сформулированы критерии оценки коммерческого веб-ресурса, объединенные в семь классов, согласно их смысловой направленности.

Ключевые слова: оценка веб-ресурса, оценка сайта, анализ сайта, критерии оценки.

Требования, предъявляемые к веб-ресурсу, с каждым годом возрастают, и наличие только необходимого контента уже недостаточно для успешного взаимодействия с большой группой пользователей. До сих пор не было разработано максимально подробного перечня критериев оценки веб-ресурса с их классификацией и пояснениями. Кроме того, используемый в исследовании подход к оценке веб-ресурсов базируется на оценке веб-ресурса не со стороны разработчика или владельца веб-ресурса, а преимущественно со стороны пользователя веб-ресурса, так как усложнение поисковых алгоритмов происходит только с целью улучшения качества поисковой выдачи, т.е. оправдания ожиданий пользователя результатами работы поисковой системы [1].

По результатам анализа различных источников были выявлены разносторонние требования к современному коммерческому веб-ресурсу, которые нашли свое отражение в сформулированных 73 критериях, разбитых позднее на 7 классов: контент, юзабилити, дизайн, репутация, технические характеристики, внутренняя оптимизация, бизнес-метрики. Классы расположены в порядке приоритета для пользователя, начиная с высшего приоритета.

1. Класс «Контент» рассматривает контент как текстовую составляющую веб-ресурса, однако к контенту могут быть отнесены также графические, видео- и аудиоматериалы веб-ресурса. Требования к графическим материалам веб-ресурса (как элементу дизайна) отражены в классе «Дизайн» данной классификации. Критерии, относящиеся к классу «Контент»:

- прозрачность понимая тематики данного веб-ресурса (наличие заголовка веб-ресурса, определение его направленности в первые секунды открытия страницы);
 - отсутствие ненормативной лексики, высказываний, призывающих к насилию или оскорбляющих честь и достоинство (текст должен быть толерантным и не включать в себя слова «лучший», «первый среди ...» без ссылок на исследования [2]);
 - высокий уровень информативности (четкое и лаконичное изложение);
 - отсутствие фактических ошибок (только достоверная информация);
 - отсутствие орфографических, грамматических и стилистических ошибок (проверить текст на наличие подобных ошибок можно на <https://glvrd.ru>);
 - структурированность текста (грамотное разделение на абзацы);
 - наличие ссылок на автора (в конце авторского материала);
 - «живость» веб-ресурса (контент должен регулярно обновляться и расширяться).
2. Класс «Юзабилити» включает критерии, отвечающие за удобство и простоту использования веб-ресурса:
- наличие навигации (меню на каждой странице);
 - отражение структуры веб-ресурса через элементы навигации (элементы навигации как основные разделы веб-ресурса);
 - использование не более 3-х кликов для поиска нужной информации (усредненное количество кликов для перехода – 3 клика [3]);
 - единообразие расположения навигации на всех страницах (обеспечивает простоту понимания и использования);
 - наличие функции Поиска (для увеличения скорости взаимодействия с пользователем);
 - отсутствие «тупиковых» страниц (возможность возвращения на предыдущую страницу без использования средств браузера);
 - обеспечение логичности ссылок (правильность перехода по ссылке);
 - отсутствие «битых» ссылок (проверять их работоспособность);
 - наличие обратной связи или возможности оставить комментарий (возможность связаться с представителями веб-ресурса);
 - доступный алгоритм покупки (в случае интернет-магазина оценка легкости пути совершения покупки, начиная с главной страницы веб-ресурса);
 - быстрый доступ к контактным данным (переход к контактным данным преимущественно в один клик [3]);
 - привычное оформление иконок (для иконок соц. сетей сохранить их узнаваемость и значение);
 - удобный для чтения размер шрифта (для desktop-версии шрифт размером от 16px [3]);
 - наличие ссылки в логотипе веб-ресурса (для быстрого перехода к главной странице веб-ресурса);
 - наличие возможности масштабирования страницы (необходимо пользователям с зрительными нарушениями [1]);
 - наличие возможности навигации по странице с использованием tab (необходимо пользователям в силу различных заболеваний [1]);
 - обеспечение доступа иностранным пользователям (контент на языке основной аудитории + английский [3]);
 - наличие текстового пояснения для сложных ситуаций навигации (наличие всплывающих подсказок для пользователя);
 - возможность связи с технической поддержкой веб-ресурса (при совершении денежных переводов, создании личного кабинета и др.);
 - возможность поделиться материалом сайта в соц. сетях (ключевая особенность поведения современных пользователей [3]).
3. Класс «Дизайн» отражает критерии, предъявляемые к внешнему виду веб-ресурса:

- уместность дизайна, его соразмерность тематике веб-ресурса (соблюдение тематической направленности);
 - цветовая палитра, не утомляющая глаза (не более двух ярких цветов для расставления акцентов);
 - наличие единого стиля на всех страницах веб-ресурса (повышает доверие пользователя и улучшает восприятие информации);
 - выравнивание элементов (более легкое восприятие контента);
 - отсутствие «тяжелых» элементов (нет агрессивной GIF-анимации);
 - отсутствие автоматического воспроизведения звука или видео (без соответствующих целевых действий пользователя);
 - качественная отрисовка элементов (в том числе использование SVG);
 - использование графических материалов высокого качества (отсутствие разбиения на пиксели);
 - использование уместного шрифта (соответствие семейства шрифтов и тематики);
 - не более двух разных шрифтов на странице веб-ресурса (для сохранения единообразия и улучшения восприятия текста);
 - отсутствие «хвостатых» ссылок (сократить ссылку – <https://goo.gl>);
 - уникальный дизайн страницы 404 (популярная тенденция [3]).
4. К классу «Репутация» отнесены критерии, отражающие доверие или недоверие пользователей к веб-ресурсу:
- возраст веб-ресурса (доверие пропорционально возрасту, если нет негативного опыта);
 - репутация веб-ресурса в глазах пользователей (личная история взаимодействия пользователя + «сарафанное радио»);
 - репутация владельцев веб-ресурса (при их публичности, личные успехи и неудачи напрямую отражаются на веб-ресурсе [3]);
 - отсутствие блоков чужой рекламы (повышает доверие пользователя [1]);
 - отсутствие автоматического скачивания вирусных программ (при загрузке веб-страницы или внутренних переходов);
 - отсутствие мошенничества (защита от злоумышленников [1]. Для проверки репутации сайта с точки зрения зараженности вирусами (наличия вредоносного кода) разработаны различные онлайн-сервисы, один из которых – www.seolik.ru/web-of-trust).
5. Также необходимо помнить и учитывать технические характеристики веб-ресурса, которые предъявляются к безопасности соединений, скорости работы, корректности отображения в различных браузерах и их версиях, в различных разрешениях экрана и др. Критерии класса «Технические характеристики»:
- охрана персональных данных пользователя (обеспечение безопасности хранения и передачи данных);
 - обеспечение https-соединения при его целевой необходимости (проверять наличие сертификата, его обновление);
 - приемлемое время ожидания загрузки страницы веб-ресурса (приемлемое время ожидания загрузки страницы – 3 с [1]);
 - корректность отображения веб-ресурса в различных браузерах и их версиях (отсутствие «поехавшей» верстки в Google Chrome, Opera, Safari, Internet Explorer, FireFox [3]);
 - адаптивность веб-ресурса (версии для desktop, mobile, планшета и широкоформатного монитора [1]);
 - отсутствие ошибок работы (бесперебойная работа веб-ресурса);
 - отсутствие дублей страниц (нет двух страниц одного раздела);
 - стабильность доступа (своевременное продление услуг хостинг-провайдера);
 - настройка «зеркал» веб-ресурса (настройки похожих url-адресов);
 - логичный адрес веб-ресурса (отражает идею веб-ресурса).

6. Критерии, относящиеся к классу «Внутренняя оптимизация», не так просто проверить обычному пользователю веб-ресурса, однако соответствие данным критериям делает веб-ресурс более доступным для пользователей за счет повышения его позиций в поисковой выдаче:
- открытие веб-ресурса поисковым роботам (для появления веб-ресурса в поисковой выдаче);
 - наличие заголовка (используя теги h1) и заголовков других уровней (для идентификации тематики веб-ресурса и его структуры);
 - наличие тега title (отображение в названии вкладки);
 - наличие ключевых слов на странице веб-ресурса (включение веб-ресурса в нужную поисковую группу [4]);
 - наличие alt-подписей в информационных иллюстрациях (полезно при проблеме с загрузкой, при поиске по картинкам);
 - наличие карты веб-ресурса (упрощает индексацию веб-ресурса).
7. К классу «Бизнес-метрики» отнесены критерии, отражающие в себе характеристики веб-ресурса, часто называемые в литературе «внешней оптимизацией веб-ресурса»:
- конверсия (показатели продаж для бизнеса);
 - лиды (целевое действие пользователя);
 - возвращения пользователя на веб-ресурс (чем больше вернувшихся, тем более успешен веб-ресурс [3]);
 - среднее число просмотров на пользователя (один просмотр на одного посетителя – плохо для веб-ресурса);
 - время, проведенное на отдельной странице веб-ресурса (определение пользователей, которые лишь бегло просматривают контент или целиком читают статью);
 - время, проведенное в целом на веб-ресурсе (пребывание в 5 мин лучше, чем в 30 с [3]);
 - показатель отказов (доля визитов с одним просмотром);
 - показатель прерываний в корзине/форме (доля посетителей, перешедших в корзину, но не сделавших заказ [3]);
 - карты кликов или «тепловые карты» (кликают ли пользователи туда, куда бы им необходимо или нет);
 - частота посещений веб-ресурса поисковым роботом (если индексный робот посещает ежедневно, значит все хорошо [3]);
 - количество проиндексированных страниц и др. (чем больше проиндексированных страниц, тем лучше для продвижения в поисковой выдаче).

Разработанная классификация может служить основой для разработки методики оценки коммерческих веб-ресурсов согласно выявленным критериям, а также базой для улучшения веб-ресурса по «западающим» показателям в целях повышения его позиций поисковой выдачи в дальнейшем.

Литература

1. Критерии оценки сайта: снимите розовые очки [Электронный ресурс]. – Режим доступа: <https://in-scale.ru/blog/kriterii-ocenki-sajta.html>, своб.
2. Федеральный закон от 13.03.2006 № 38-ФЗ (ред. от 29.07.2017) «О рекламе» [Электронный ресурс]. – Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_58968, своб.
3. Критерии оценки сайта [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/120989/>, своб.
4. Маматов Е.М., Брусенская И.Н. Сравнительная оценка влияния факторов продвижения сайта в поисковых системах Яндекс и Google // Научные ведомости белгородского государственного университета. – 2015. – № 13-1. – С. 109–113.

**Маревская Мария Владимировна**

Год рождения: 1989

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: marevskaiamaria@gmail.com

**Государев Илья Борисович**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: goss@corp.ifmo.ru

УДК 004

**ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ВЕБ-ФРЕЙМВОРКОВ ПРИ РАЗРАБОТКЕ
ДЕСКТОПНЫХ ПРИЛОЖЕНИЙ****Маревская М.В., Государев И.Б.****Научный руководитель – к.ф.-м.н. Соловьев И.А.**

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

В работе проведен обзор фреймворков, позволяющих создавать десктопные приложения на основе веб-технологий, рассмотрены фреймворки Electron и NW.js, их общие черты, специфика их использования, различия между ними.

Ключевые слова: веб-фреймворк, Electron, NW.js, десктопное приложение, кроссплатформенное приложение.

Веб-фреймворки Electron и NW.js позволяют создавать десктопные приложения с помощью стандартного стека веб-технологий, используя HTML, CSS, JavaScript. Они основаны на платформе Node.js, браузере Chromium и движке интерпретации JavaScript V8. С помощью специальных библиотек и благодаря специфике Node.js можно получить доступ к файловой системе, буферу обмена, камере, обрабатывать системные запросы, – тем возможностям, которые являются обязательными для любого десктопного приложения, созданного с помощью специфического IDE и «родного» языка для каждой платформы. Такое приложение может открываться в полноэкранном режиме, сворачиваться в трей, включать в себя системное меню, состоять из нескольких окон и элементов интерфейса, имеет доступ к кросс-доменным запросам.

NW.js и Electron позволяют создавать кроссплатформенные приложения. Для создания приложения, работающего под Windows, MacOS или Linux, необходимо выбрать соответствующий вариант компиляции. Приложения легко упаковываются и распространяются [1, 2].

Несмотря на то, что разработка на Electron и NW.js похожа на веб-разработку и доступна для веб-программиста, работающего на Node.js, она имеет свою специфику.

1. NW.js. NW.js – проект, ранее называвшийся node-webkit, спонсорами выступают компании Intel и Gnor Technologies. Разработка была начата в 2011 г. [3]. Поддерживается Node.js API и сторонние модули. Все модули Node.js могут быть напрямую вызваны из DOM [4].

NW.js использует различный JavaScript контекст для каждого окна и Node.js контекст. API NW.js загружаются в объект nw и могут быть использованы в файлах JavaScript [2].

2. Electron. Electron – open source проект, был создан в 2013 г. компанией Github как платформа для редактора Atom, изначально назывался Atom Shell. В мае 2014 г. были открыты исходники, в апреле 2015 проект был переименован в Electron. С мая 2016 г. приложения совместимы с Mac App Store, с августа 2016 – с Windows Store. Electron объединяет Chromium и Node.js в единую среду выполнения [1].

В Electron существует два вида процессов – основной процесс и процесс визуализации. Основной процесс может быть только один, он отвечает за взаимодействие с операционной системой и создает процессы визуализации – отображение окна браузера [1].

Электрон использует Node API, предоставляемый Node.js, Web API, предоставляемый Chromium, и свой собственный API. Для доступа к API Electron необходимо импортировать модуль electron:

```
const electron = require('electron')
```

Модуль BrowserWindow позволяет создавать новое окно браузера или управлять уже существующим. Каждое окно браузера является отдельным процессом визуализации. Этот процесс, как и главный процесс, контролирующей жизненный цикл приложения, имеет полный доступ к Node.js API.

Для создания меню используются модули Menu и MenuItem. Если меню не задано, Electron сгенерирует минимальное меню самостоятельно. Эти же модули используются для создания контекстного меню.

Модуль shell предоставляет доступ к некоторым нативным возможностям, таким как браузер по умолчанию и файловый менеджер. Этот модуль используется также для открытия ссылок не внутри приложения, а вовне, в браузере, который используется по умолчанию.

Модуль dialog позволяет использовать нативный системный диалог для открытия файлов и папок, сохранения файлов и отображения информационных сообщений.

К некоторым API можно получить доступ только из основного процесса, к некоторым из основного и из процесса рендеринга. Если необходимо вызвать API, доступное только из основного процесса, это можно сделать с помощью межпроцессорного взаимодействия через основной процесс [1].

Необходимо заметить, что при создании кроссплатформенных приложений интерфейса для различных операционных систем необходимо руководствоваться соответствующими руководствами для каждой системы.

Несмотря на то, что фреймворки NW.js и Electron имеют много общего, между ними есть некоторые различия:

- Electron более походит на среду выполнения Node.js, поэтому точкой входа в приложение является только JavaScript-сценарий, из которого вручную нужно загружать страницу в окно браузера и прослушивать события окна для корректного выхода из приложения, тогда как в NW.js точкой входа может быть и HTML-страница;
- в Electron используется только часть Chromium – библиотека рендеринга libchromiumcontent, что позволяет избежать сборки всей библиотеки Chromium;
- разница в реализации событийного цикла (event loop);
- использование веб-контекста и контекста Node.js в NW.js (начиная с версии 0.13 NW.js также поддерживает смешанный контекст) и использование смешанного контекста в Electron [1].

Таким образом, фреймворки NW.js и Electron предоставляют сходные возможности при разработке десктопных приложений, используют концепции и подходы веб-программирования. Созданное приложение сочетает в себе как элементы системного интерфейса, так и элементы веб-страниц. Благодаря использованию языка и технологии веб-разработки рассмотренные фреймворки позволяют сделать разработку десктопных приложений доступной для веб-программистов.

Литература

1. Документация Electron [Электронный ресурс]. – Режим доступа: <https://electronjs.org/docs/all> (дата обращения: 05.01.2018).
2. NW.js Documentation for 0.13 and later [Электронный ресурс]. – Режим доступа: <http://docs.nwjs.io/en/latest/> (дата обращения: 05.01.2018).
3. Node-webkit is renamed NW.js [Электронный ресурс]. – Режим доступа: <https://github.com/nwjs/nw.js> (дата обращения: 05.01.2018).
4. What you should know before starting node-webkit [Электронный ресурс]. – Режим доступа: <http://yedingding.com/2014/08/01/node-webkit-intro-en.html> (дата обращения: 05.01.2018).



Мартюшов Михаил Валентинович

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: martyushovmv@gmail.com



Шуклин Дмитрий Анатольевич

Год рождения: 1979

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: do@limtu.ru

УДК 004.738.5

**ПЕРСПЕКТИВНЫЕ НАПРАВЛЕНИЯ РАЗВИТИЯ ВСПОМОГАТЕЛЬНЫХ
СЕРВИСОВ ДЛЯ ПРОДВИЖЕНИЯ ВЕБ-САЙТОВ**

Мартюшов М.В., Шуклин Д.А.

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 рассмотрены разновидности вспомогательных сервисов для продвижения веб-сайтов, а также главные функции этих инструментов, определены основные направления развития данной сферы, сделаны соответствующие выводы.

Ключевые слова: Search Engine Optimization (SEO), продвижение сайтов, поисковые системы, сервисы для продвижения, оптимизация, SEO-инструменты.

Стремительное развитие веб-технологий способствовало появлению большого количества коммерческих сайтов и сервисов в сети Интернет. Бизнес все больше и больше переходит в веб-среду. В связи с этим за последнее время сильно возросла потребность в продвижении коммерческих интернет-ресурсов. Это не могло не отразиться на сфере поисковой оптимизации веб-сайтов, которая за последние несколько лет значительно преобразилась и продолжает меняться до сих пор.

Продвижение интернет-ресурсов стало сложнее: изменились алгоритмы поисковых систем, появились совершенно новые методы продвижения, а конкуренция стала еще жестче. Кроме того, в поисковых системах появились специальные «фильтры», попасть под которые рискуют те, кто использует недобросовестные или не совсем честные методы продвижения.

Чтобы хоть немного упростить и ускорить процесс поисковой оптимизации создаются вспомогательные сервисы для продвижения веб-сайтов. На данный момент существует большое количество как платных, так и бесплатных вспомогательных SEO-сервисов различной направленности. Каждый из них обладает определенным функционалом, который зависит от назначения и сложности сервиса. Простые сервисы, как правило, выполняют одну конкретную функцию, либо небольшую группу однородных функций. Крупные сервисы, в свою очередь, представляют собой целые многофункциональные системы, которые значительно упрощают продвижение сайта. Круг пользователей вспомогательных SEO-сервисов определяется как владельцами сайтов, продвигающими свой интернет-ресурс, так и работниками сферы продвижения.

Целью настоящей работы являлось определение основных сфер применения вспомогательных сервисов для продвижения веб-сайтов, а также выявление наиболее перспективных направлений развития подобных сервисов.

Основные группы SEO-сервисов. Все вспомогательные сервисы для продвижения веб-сайтов можно условно разделить на группы, определив тем самым классификацию таких сервисов. Она состоит из пяти групп:

1. инструменты поисковых систем. В данную группу входят вспомогательные сервисы продвижения, выпускаемые и поддерживаемые самими поисковыми системами. Это, прежде всего, сервисы от ведущих компаний: Яндекс и Google. Сюда относятся панели управления и настроек Яндекс.Вебмастер и Google Webmaster Tools, инструменты оценки различных SEO-метрик Яндекс.Метрика и Google Analytics, которые позволяют узнать множество важных характеристик сайта, а также информацию о том, как он определяется поисковыми системами [1]. Сервисы подбора ключевых слов Яндекс.Вордстат и Google Trends, позволяющие оценить статистику частотности по ключевым словам [2]. Также сюда можно отнести сервисы размещения контекстной рекламы Яндекс.Директ и Google Adwords. Следует отметить, что сервисы конкретной поисковой системы разработаны специально для облегчения процесса продвижения именно в этой поисковой системе;
2. аналитические сервисы. Инструменты этой группы обеспечивают различную информационную поддержку в продвижении интернет-ресурсов. Среди них можно выделить отдельные категории сервисов по выполняемым функциям. Сервисы SEO-анализа сайтов позволяют детально анализировать как страницы личного сайта, так и сайты конкурентов. Сервисы по определению позиции сайта в поисковой выдаче предназначены для того, чтобы проверять, на каком месте в списке результатов поиска находится веб-сайт при определенном запросе в конкретной поисковой системе, а также следить за динамикой изменений. Сервисы анализа обратных ссылок позволяют оценить количество и эффективность ссылок, ведущих на сайт. Благодаря сервисам анализа контента можно определять степень плотности размещения ключевых слов, заспамленность и академическую тошноту текстов, их уникальность и прочие параметры;
3. сервисы автоматизации продвижения. Данную группу инструментов составляют онлайн-приложения, продвигающие ресурсы в сети Интернет путем автоматической закупки обратных ссылок. Такие сервисы были очень востребованы, когда ссылочный фактор оказывал существенное влияние на позицию сайта в поисковой выдаче. В настоящее время значение этого фактора уменьшилось в несколько раз. Соответственно, актуальность сервисов автоматизации продвижения также снизилась. Однако их продолжают использовать в качестве вспомогательного инструмента. Что касается набора функций таких сервисов, то в него входят анализ релевантности страниц продвигаемым запросам, автозакупка ссылок со всеми необходимыми параметрами, составление анкоров ссылок на полном автомате, накрутка поведенческих факторов, автоматическое отслеживание изменения позиций сайта в выдаче поисковиков. Яркими примерами сервисов автоматического продвижения являются SeoPult, Rookee, WebEffector и MegaIndex;
4. биржи продвижения интернет-ресурсов. Эти сервисы представляют собой веб-базированные биржи, которые предназначены для создания конкретных заданий по продвижению интернет-ресурсов для их реализации многочисленными исполнителями. Причем задания могут быть самыми разными: от простого посещения целевого сайта до написания полноценных отзывов на других ресурсах с упоминанием названия или прямой ссылкой на веб-сайт. Примерами таких бирж являются порталы qcomment.ru, forumok.ru, advego.ru и другие;
5. стратегические SEO-инструменты. Сюда относятся системы поддержки принятия решений и подобные им сервисы. Это совершенно новый вид инструментов в сфере продвижения веб-сайтов. К сожалению, в силу специфических особенностей развития

направления SEO-оптимизации, разработка подобных систем и сервисов до недавнего времени не интересовала исследователей, и на данный момент нет готовых проектов такого типа. Однако авторы (S. Sagot, A-J. Fougères, E. Ostrosi) затронули данную тему в своих научных работах [3, 4], и в итоге пришли к тому, что создали рабочий прототип системы поддержки принятия решений в сфере SEO. Системы подобного типа позволят владельцам сайтов лучше разбираться в том, как работают методы продвижения, а также помогут понять, что они делают правильно, а что нет.

Проблемы и перспективы. Современные вспомогательные SEO-сервисы разных типов и назначений активно используются как профессионалами, так и новичками. Спрос есть и на платные, и на бесплатные решения. Большой интерес у пользователей вызывают многофункциональные сервисы, поэтому прослеживается четкая тенденция создания крупных веб-базируемых порталов, объединяющих в себе целый ряд SEO-инструментов. По большей части это сервисы, в которых собраны вместе различные инструменты SEO-аналитики.

Инструменты поисковых систем постепенно улучшаются и приобретают новые функции. Эти сервисы очень востребованы и не теряют своей актуальности, но далеко не все пользователи знают про них или умеют ими пользоваться, поэтому зачастую используют сторонние интернет-ресурсы.

Сервисы автоматического продвижения стали менее востребованными после значительного уменьшения влияния ссылочного фактора на позицию веб-сайта в поисковой выдаче, а также после введения в действие поисковыми системами новых фильтров. Однако разработчики делают все возможное, чтобы вернуть популярность этим инструментам, усложняя алгоритмы работы и добавляя новые функции, автоматизирующие различные процессы при продвижении сайта. Стоит отметить, что сама возможность автоматизации большого количества однотипных действий делает это направление перспективным в плане дальнейшего развития.

Активно развиваются биржи продвижения, позволяющие значительно упростить и ускорить продвижение веб-сайта не хуже сервисов автоматизации продвижения. Такие биржи привлекают большое количество и заказчиков, и исполнителей, поскольку и тем и другим выгодно такое сотрудничество. Очевидно, что разнообразные формы подобных сервисов будут и дальше появляться на рынке, охватывая все больше и больше методов продвижения интернет-ресурсов.

Довольно перспективным направлением является создание систем поддержки принятия решений для SEO и подобных им сервисов. Несмотря на отсутствие готовых решений в данной сфере, интерес к этой теме есть, и с большой долей вероятности можно сказать, что такие продукты будут пользоваться спросом на рынке. Наиболее возможными представляются реализации таких сервисов в форме самих систем поддержки принятия решений, а также систем разработки стратегии продвижения и рекомендательных SEO-систем.

Литература

1. Исаев А.Р. 9 простых шагов по SEO-оптимизации вашего сайта // Интернетнаука. – 2017. – № 7. – С. 21–24.
2. Хафизов Д.Г., Игнатенко М.В., Сазанова И.Н., Ухов С.В. Продвижение сайта в интернет: основы, сервисы и инструменты // Актуальные направления научных исследований XXI века: теория и практика. – 2015. – № 8-1. – С. 372–375.
3. Sagot S., Fougères A-J., Ostrosi E. A multi-agent approach for building a fuzzy decision support system to assist the SEO process // IEEE International Conference on Systems, Man, and Cybernetics. – 2016. – P. 4001–4006.
4. Sagot S., Fougères A-J., Ostrosi E. Business Constraints Integration in a Search Engine Optimization Fuzzy Decision Support System // IEEE 19th Conference on Business Informatics. – 2017. – P. 302–311.

**Матушевский Никита Евгеньевич**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: nickmat95@yandex.ru

УДК 005

**ОПРЕДЕЛЕНИЕ ОПТИМАЛЬНОГО АЛГОРИТМА НЕЧЕТКОГО ПОИСКА
ДЛЯ НАХОЖДЕНИЯ ДУБЛИКАТОВ КАРТОЧЕК ПАЦИЕНТОВ В МЕДИЦИНСКОЙ
ИНФОРМАЦИОННОЙ СИСТЕМЕ****Матушевский Н.Е.****Научный руководитель – д.т.н., доцент В.И. Погорелов**

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

В работе приведен краткий обзор методов нечеткого поиска текста по словарю, их анализ по определенным критериям, для решения задачи составления алгоритма поиска дубликатов карточек пациентов в медицинской информационной системе, а также выбор наиболее оптимального алгоритма.

Ключевые слова: медицинская информационная система, МИС, нечеткий поиск, алгоритмы нечеткого поиска по словарю, дубликат карточки пациента.

В работе [1] были рассмотрены проблемы современных медицинских информационных систем и выявлена наиболее значимая – наличие дубликатов карточек пациентов, а также, предложен вариант ее решения.

Дубликат карточки пациента – карточка, заведенная на пациента, уже имеющего карточку в медицинской информационной системе, иначе говоря, повторная регистрация пациента.

Проблему было принято решить путем разработки интегрируемого модуля поиска и объединения дубликатов карточек пациентов. Процесс поиска дубликатов – наиболее важная и ресурсозатратная процедура, поэтому для реализации поиска необходимо было разработать алгоритм, выдающий наибольшее количество совпадений за наименьшее количество времени.

Для этого было принято решение, реализовывать алгоритм на основании алгоритма нечеткого поиска по словарю.

В данной работе был проведен сравнительный анализ алгоритмов нечеткого поиска по словарю, и выбран наиболее оптимальный для решения выше обозначенной проблемы.

Основной характеристикой алгоритма нечеткого поиска является метрика.

Метрика предназначена для определения разницы между искомым и сравниваемым словом.

Разница между словами выражается числом, выражающим, на сколько символов одно слово отличается от другого. Также разницу между словами называют расстоянием.

Рассмотрим следующие виды метрик.

1. Расстояние Хемминга – метрика, определяющая расстояние между словами одинаковой длины.

Пример: сравним слова «борец» и «боров». Данные слова отличаются на 2 символа, соответственно, расстояние Хемминга будет равняться 2.

Недостатки:

- способен определять расстояние только между словами одинаковой длины.

2. Расстояние Левенштейна – минимальное количество операций, необходимых для преобразования одной строки в другую.

Для преобразования используются операции добавления, удаления и замены символов.

Пример: сравним слова «собака» и «облака». Расстояние Левенштейна между этими словами равно 3, так как самый короткий способ получить из одного слова другое – использовать операцию замены для первых трех букв.

Недостатки:

- при перестановке местами частей слов, расстояние между сравниваемыми словами оказывается значительно больше;
- расстояние между длинными, но достаточно похожими словами, оказывается большим, тем временем как расстояние между совершенно разными короткими словами остается небольшим [2].

3. Расстояние Дамерау–Левенштейна, является модификацией для алгоритма расстояния Левенштейна. Модификация заключается в добавлении еще одного метода преобразования – транспозиции двух соседних букв.

Данная модификация позволяет увеличить точность рассчитываемого расстояния, относительно алгоритма расстояния Левенштейна.

Для выбора алгоритма метрики, были выдвинуты следующие требования:

- алгоритм должен производить расчет расстояния для слов разной длины;
- алгоритм должен давать максимально точный результат при сравнении слов разной длины.

На основании выдвинутых требований был выбран алгоритм расчета метрики Дамерау–Левенштейна, так как, являясь модификацией алгоритма расчета метрики Левенштейна, он дает более точный результат, а алгоритм Хемминга не удовлетворяет условию первого требования.

Далее, исходя из результатов исследования автора источника [3], было выбрано три алгоритма нечеткого поиска, показавших наибольшие результаты:

1. алгоритм ВК-деревьев, так как дает один из наибольших процентов полноты результата;
2. метод расширенной выборки, так как дает максимальный процент полноты результата;
3. метод N -грамм (модификация 2), так как дает один из наибольших процентов полноты результата, а также имеет максимальную скорость работы.

Все вышеперечисленные алгоритмы ведут работу со словарями.

Под словарем, в данном контексте, понимается некий набор слов, с которыми будет производиться сравнение. При решении задачи поиска опечаток, словарь должен быть составлен из грамматически верных слов.

Идеальный словарь должен состоять из всех слов данного языка, но в действительности словари формируют исключительно под решаемые задачи. Если говорить о поиске опечаток, к примеру, в английском языке, то средний словарь будет содержать около 62 000 базовых слов [4].

Для выбора оптимального алгоритма был создан словарь из русских имен и отчеств.

Далее необходимо реализовать алгоритм и провести исследование, заключающееся в следующем:

1. ввести поочередно неверно 50 слов, имеющих в словаре;
2. измерить время поиска для каждого слова;
3. измерить количество верно найденных слов;
4. на основании исследования № 2, вычислить время выполнения поиска для каждого слова из базы данных, размером 15 000 записей.

Алгоритм ВК-деревьев. ВК-дерево – это метрическое дерево, разработанное Уолтером Остином Буркхардом и Робертом М. Келлером [5].

Метрическое дерево – это любая древовидная структура данных, специализирующаяся на индексировании данных в метрических пространствах [6].

Результаты тестирования:

- время выполнения поиска для одного слова: 0,4 с;
- количество найденных слов: 100%;
- среднее время поиска для базы данных: около 90 мин.

Метод N -грамм – метод, основывающийся на следующем принципе: «модифицированное и исходное слова должны обладать общими подстроками» [7].

N -грамма – это последовательность из n -элементов [8].

Суть алгоритма заключается в разбиении искомого слова на n -граммы, а затем в проверке на наличии n -грамм подстрок в слове, с которым ведется сравнение. Чем больше вхождений, тем больше вероятность, что это нужное нам слово.

Слово может быть разбито на любое количество n -грамм, но оптимальным принято считать разбиение на триграммы [9].

Результаты тестирования:

- время выполнения поиска для одного слова: 0,2 с;
- количество найденных слов: 92%;
- среднее время поиска для базы данных: около 45 мин.

Алгоритм расширения выборки – алгоритм, приводящий задачу нечеткого поиска, к задаче четкого поиска.

Алгоритм используется в тех случаях, если размер словаря не велик или же, если время выполнения поиска не играет особого значения [3].

Принцип работы алгоритма заключается в следующем: из словаря корректно написанных слов составляется словарь слов со всевозможными ошибками, далее входное слово сравнивается со всеми ошибочными вариантами, и при совпадении возвращается верный вариант написания ошибочного варианта.

Данный алгоритм является самым простым в плане логики работы и реализации.

Преимуществом данного алгоритма является наивысший результат поиска [3].

Однако у него есть существенный недостаток. Время работы алгоритма сильно зависит от предполагаемого количества ошибок. Таким образом, при увеличении количества ошибок, количество запросов к сгенерированному словарю может составлять десятки тысяч, что существенно увеличивает время работы.

Исходя из того, что рассмотренные выше алгоритмы показывают высокие показатели результативности за приемлемый промежуток времени, данный алгоритм реализован не был.

После проведения исследования были сравнены результаты для алгоритма ВК-деревьев и метода N -грамм.

Метод N -грамм выигрывает по временному показателю у алгоритма ВК-деревьев, однако, проигрывает по количеству найденных слов.

Временной фактор является менее приоритетным, так как работа модуля предполагается в фоновом режиме.

Исходя из этого, для дальнейшей реализации был выбран алгоритм ВК-деревьев.

Литература

1. Матушевский Н.Е. Определение и анализ недостатков, существующих МИС // Альманах научных работ молодых ученых Университета ИТМО. – 2017. – Т. 4. – С. 161.
2. Расстояние Левенштейна [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Расстояние_Левенштейна, своб.
3. Сметанин Н. Нечеткий поиск в тексте и словаре [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/114997/>, своб.
4. Spelling check algorithms [Электронный ресурс]. – Режим доступа: <https://cs.brynmawr.edu/Courses/cs330/spring2012/SpellingCheckers.pdf>, своб.

5. ВК-Tree [Электронный ресурс]. – Режим доступа: <https://en.wikipedia.org/wiki/ВК-tree>, своб.
6. Metric tree [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Metric_tree, своб.
7. Краткий обзор методов поиска ключевого слова по словарю [Электронный ресурс]. – Режим доступа: <http://itpool.ru/blog/kratkiy-obzor-metodov-poiska-klyuchevogo-slova-po-slovaryu.html>, своб.
8. N-грамма [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/N-грамма>, своб.
9. Токарь А. Нечеткий поиск в словарях фиксированного размера [Электронный ресурс]. – Режим доступа: <https://www.slideshare.net/alexeytokar9/ss-49357090>, своб.

**Миннемуллин Булат Мансурович**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4107Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: b@minbulat.ru

**Погорелов Виктор Иванович**

Год рождения: 1940

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, д.т.н., профессор

e-mail: kpd@limtu.ru

УДК 004.588**ИГРОФИКАЦИЯ В ДИСТАНЦИОННОМ ОБУЧЕНИИ ШКОЛЬНИКОВ****Миннемуллин Б.М., Погорелов В.И.****Научный руководитель – д.т.н., профессор Погорелов В.И.**

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

В работе показаны результаты анализа исследовательских работ в сфере игрофикации и ее применения в дистанционном обучении. Рассмотрены примеры приложений, использующих игрофикацию при обучении школьников, и ресурсы, позволяющие создавать игрофицированные курсы.

Ключевые слова: игрофикация, дистанционное обучение, общее образование, обучающие приложения.

Дистанционное образование получило широкое распространение в России и в мире. В последние годы появилось большое количество платформ, организующие массовые обучающие онлайн-курсы, во многих университетах появляются программы, позволяющие полностью получить образование, не выходя из дома. Существует много похожих понятий: дистанционное обучение, электронное обучение, онлайн-обучение и другие. Значения этих понятий отличаются, но во многих источниках они применяются в искаженном значении. В данной работе под дистанционным обучением понимается организация образовательной деятельности, при которой взаимодействие между участниками образовательного процесса происходит опосредованно через компьютерные сети.

Дистанционное обучение имеет много преимуществ и недостатков. Оно позволяет обучаться из разных уголков планеты, что дает равные возможности для разных социальных слоев населения независимо от места проживания, состояния здоровья, национальности и материального состояния. Дистанционное обучение создает возможность изучения дисциплин в индивидуальном темпе, самостоятельного расчета времени и продолжительности занятий, доступность обучения для любого человека, возможность выбора многочисленных курсов обучения. К преимуществам также можно отнести возможность повторного изучения курсов для обновления знаний. Основной недостаток, не позволяющий полностью перейти в школьном образовании на дистанционное обучение – это

отсутствие реального общения между учениками и преподавателями. Кроме этого, необходимость хорошей технической оснащённости дома для получения образовательных материалов не позволяет получать дистанционное образование бедным слоям населения. Также не всегда электронные курсы хорошо проработаны, часто обучение ведется в письменном виде, не предоставляя возможности обучающимся излагать свои мысли в устной форме, что может повлечь некачественное усвоение информации. Сложность организации самодисциплины не позволяет обучающимся подолгу пользоваться дистанционными образовательными ресурсами [1]. Для устранения последнего недостатка предлагается использовать игрофикацию.

Игрофикация – это применение игровых элементов и игровых механик в неигровом контексте. Игрофикации применяется во многих областях: маркетинге, управлении персоналом, образовании, защите окружающей среды.

Применяемые в игрофикации элементы разделяются на три категории: динамики, механики и компоненты.

Динамики – это общие аспекты игрофицированной системы. Их невозможно непосредственно внедрить в работу, но нужно принимать во внимание.

К ним относятся:

- ограничения (лимиты или вынужденные компромиссы);
- эмоции (любезность, дух соперничества, разочарование, счастье);
- повествование (последовательная, непрерывная сюжетная линия);
- продвижение (рост игрока и его развитие);
- отношения (социальные взаимодействия, формирующие чувства товарищества, статуса и альтруизма).

Механики – это процессы, которые движут действиями игрока и формируют у него вовлеченность. Это очки, бейджи (значки достижений), уровни, таблицы лидеров.

Компоненты – это конкретная форма, которую принимают механики и динамики. Например, очки можно разделить на очки опыта, деньги, репутация, умения и карма.

Применение игрофикации в образовании позволяет помочь легко сфокусироваться, преобразовать тяжелую работу в более легкие задачи, увеличить мотивацию и участие обучающихся, помогает им быть более активными и сознательными при удовлетворении своих потребностей в обучении, избавить от скуки.

Игровые технологии в образовании начали применять давно, и поэтому они хорошо проработаны как российскими, так и зарубежными исследователями. О роли игры в воспитании и обучении в своих трудах писал Ушинский и Делли.

Элементы игрофикации применяются в школах и в настоящее время. Например, оценки, которые ставят в школе, можно отнести к механике «Очки», а переход с одного класса в другой – это механика «Уровни».

Вопросы применения игрофикации в разных сферах жизни неоднократно рассматривались в работах кафедры КПиД Университета ИТМО. В работах показываются использование игрофикации для повышения мотивации студентов при дистанционном обучении [2], применении структурной игрофикации [3], а также реализация игрофикации на различных платформах [4].

Эффективность применения средств игрофикации исследуется многими авторами. В [5] проведен анализ статей, в которых описывались результаты использования игрофикации в образовании школьников. В большинстве из них используются очки, таблицы лидеров и бейджи. В 59% исследованиях дается частично положительный результат, отрицательных результатов нет (рисунок). Таким образом, в целом применение игрофикации положительно сказывается на процессе обучения.

Существует много образовательных проектов, направленных на дополнительное образование людей: общее развитие человека, изучение иностранных языков, подготовка к

ЕГЭ и другим экзаменам, массовые курсы и другие. Многие ресурсы применяют игрофикацию для повышения вовлеченности пользователей и удержания их внимания.

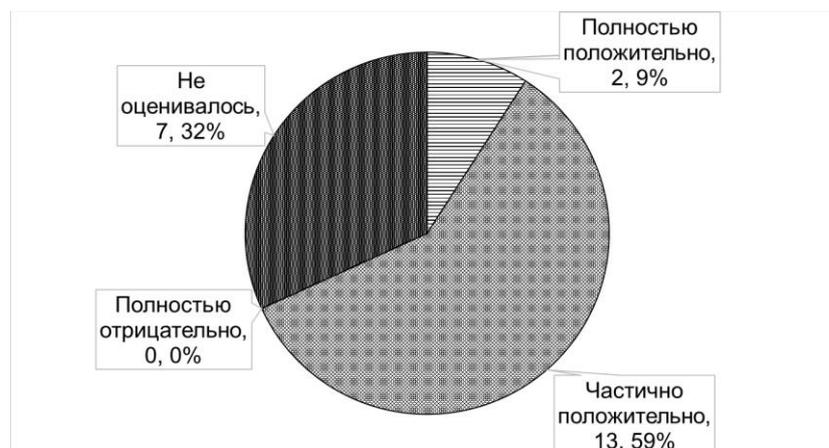


Рисунок. Оценка эффективности игрофикации в анализируемых исследованиях

Наиболее игрофицированная сфера образования – это изучение иностранных языков. В последнее время были разработаны много ресурсов: сайты, мобильные приложения, настольные компьютерные приложения – для обучения иностранным языкам.

Зарубежный ресурс Duolingo – полностью асинхронный, виртуальный курс изучения иностранных языков. Пользователю предлагается самостоятельно изучать новые слова, а после прохождения определенного времени, приложение напоминает, что слова долгое время не повторялись и могли быть забыты. В приложении применяются несколько элементов игрофикации: таблицы лидеров, очки, ежедневные задания.

Отечественный онлайн-ресурс по изучению английского языка LinguaLeo.ru также содержит элементы игрофикации. За счет внедрения игровых элементов в процесс обучения, данному ресурсу удалось завоевать огромную популярность и стать значимым явлением в образовательном сегменте. Система включает элементы социальных сетей, что делает обучающую среду более гибкой и привычной для широкого круга пользователей.

В сфере подготовки к экзаменам существует много приложений. Например, проект Examer является сайтом, где пользователь может самостоятельно готовиться к сдаче единого государственного экзамена по разным предметам. Каждому пользователю приложение строит индивидуальный план подготовки, выполняя который ученик может улучшить свои знания по разным предметам. В приложении применяются много элементов игрофикации: очки, список лидеров, бейджи. Проект показал большой рост аудитории: за год приложением начали пользоваться более 100 000 человек.

Существует множество платформ, на базе которых можно реализовать отдельные идеи по игрофикации курсов обучения.

Популярная платформа для создания электронных курсов Moodle позволяет применять механики и элементы игрофикации. В ней есть возможность использования значков, списков лидеров, баллов, индикаторы выполнения, создания дорожных и ментальных карт, на основе которых можно реализовать игрофикацию обучения.

Также существуют специальные сервисы, предназначенные для разработки дистанционных курсов.

Зарубежный сервис TalentLMS специализируется на разработке игрофицированных образовательных курсов. Приложение позволяет использовать методы игрофикации: очки, значки достижений, таблицы лидеров и другие. Все эти параметры можно настраивать.

iSpring – отечественный онлайн-сервис, позволяющий публиковать подготовленные курсы и проводить контроль знаний. В процессе обучения пользователи набирают очки при прохождении тестов, тренажеров и выполнении заданий, выдаются бейджи, строится

рейтинг обучающихся. Приложение позволяет настраивать дизайн и различные функции элементов игрофикации.

В процессе исследования выявлено, что дистанционное образование имеет множество преимуществ и недостатков. Игофикация приносит в процесс обучения состязательность, позволяет привлекать и удерживать внимание обучающихся. Описанные в работе приложения показывают возможность и эффективность игрофикации обучения.

Литература

1. Набиев И.М. Перспективы дистанционного образования // Молодой ученый. – 2014. – № 2. – С. 799–801.
2. Шуклин Д.А., Погорелов В.И., Зими́на Д.В., Козак О.О. Особенности использования элементов геймификации в системах дистанционного обучения // Технические науки. Теория и практика. Материалы II международной научно-практической конференции. – 2017. – С. 68–72.
3. Погорелов В.И., Козак О.О., Зими́на Д.В. Применение методов структурной геймификации при дистанционном обучении // Современное образование: проблемы взаимосвязи образовательных и профессиональных стандартов. Материалы международной научно-методической конференции. – 2016. – С. 106–107.
4. Зими́на Д.В., Козак О.О., Погорелов В.И., Шуклин Д.А. Применение протокола LTI для интеграции уроков STEPIK в обучающий курс на платформе Wordpress // Перспективы науки. – 2017. – С. 12–14.
5. Hamari J., Koivisto J., Sarsa H. Does Gamification Work? – A Literature Review of Empirical Studies on Gamification // 47th Hawaii International Conference on System Science. – 2014. – P. 3025–3034.

**Молодцова Виктория Сергеевна**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4105Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: VictoriaWD@yandex.ru

**Шуклин Дмитрий Анатольевич**

Год рождения: 1979

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: do@limtu.ru

УДК 331.101.5**ТЕХНОЛОГИЯ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ В АРХИТЕКТУРЕ****Молодцова В.С., Шуклин Д.А.****Научный руководитель – к.педагог.н., доцент Шуклин Д.А.**

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

В работе рассмотрено современное состояние и перспективы использования технологии дополненной реальности. Проведен обзор современных приложений для профессиональной деятельности архитекторов.

Ключевые слова: AR, дополненная реальность, архитектура, виртуальная реальность, плагин, модуль, маркер, визуализация.

Что же такое дополненная реальность? Первое, что придет в голову любому человеку – это всем известные мечты о попадании в виртуальность, полном погружении, будь то в неизведанный мир кинематографа или компьютерной игры. Однако это далеко не так.

Дополненная реальность (augmented reality, AR) – это технология представления контекстной информации и наложения ее в виде многослойных визуальных образов на объекты реального мира в режиме реального времени [1]. Дополненная реальность (ДР) является уникальной платформой нового интерфейса для перехода на новый интерактивный уровень посредством взаимодействия и обращения к информации. Отличие дополненной реальности от виртуальной заключается во взаимодействии компьютерных устройств с объектами реального мира. Главной задачей ДР является расширение информационного пространства для взаимодействия пользователя с окружением. Накладываемые компьютерным устройством слои носят вспомогательно-информативный характер контекстных объектов на изображение реальной среды [1]. Таким образом, информация, связанная с объектами реального мира, становится доступной пользователю в режиме реального времени с помощью ДР.

Стоит отметить также принцип работы приложения ДР. Видеопоток с камеры аппарата переходит в модуль наблюдения камеры. Этот модуль воспроизводит любой эпизод видеопотока: осуществляет отбор заданного предварительно маркера, устанавливает состояние маркера в месте, и на основании данных сведений исчисляет состояние условной

камеры относительно маркера. После того, как состояние и направленность камеры установлены, модуль хранения предметов размещает на сцену нужный предмет с целью визуализации, применяя предварительно выбранные характеристики положения, масштаба и поворота. Далее происходит модификация с применением сдвига согласно отношения расчетной позиции к ориентации. Пользователь способен воздействовать на характеристики модификации и визуализации с помощью пользовательского интерфейса [2].

В настоящее время самыми популярными средствами просмотра стали смартфоны и планшеты с установленными на них специальными приложениями для работы с технологией дополненной реальности. Данная технология не только завоевала внимание профессионалов разных областей науки, но также прочно вошла в повседневность использования многих людей для решения самых различных задач. Что касается архитекторов (а также дизайнеров, строителей и т.д.) – для них данная технология ДР представляет собой нечто большее, чем просто очередная инновация. AR-media способны заменить кипы чертежей и сэкономить деньги и время при строительстве [3].

Рынок мобильных приложений для архитекторов формируется в нескольких направлениях. В сфере проектов для творчества захватила лидерство фирма Morpholio. Данное приложение позволяет на изображение накладывать другие слои, своего рода имитация кальки, визуализированных объектов, позволяющие осуществлять новые решения.

Augment используется в качестве плагина в программы для создания отдельных объектов, экстерьеров и строений – данное приложение позволяет визуализировать 3D-модели в реальном времени и в натуральную величину. Пользователю нужно лишь загрузить сборники объектов в библиотеку и распечатать листы с особыми метками. Далее направляем гаджет на листок с маркером, выбираем модель из каталога – и Augment встраивает виртуальный объект в реальный интерьер (или малые архитектурные формы – во двор дома) [4].

Visidraft работает примерно так же, как и Augment, но данная программа имеет существенное отличие – не нужно использовать маркер. Достаточно загрузить 3D-модели мебели в каталог – и пользователь может расставить предметы на экране планшета по своему усмотрению. Преимуществом приложения также является способность запоминать расположение всех объектов и просматривать объекты в виртуальном туре, сидя на одном месте.

Urbasee Future способен «расширить» реальность не только определенным предметом, но и полноценным строением. Пользователь загружает макет здания на сервер, и на интерактивной карте привязывает его с помощью геолокации. Когда оказывается на нужном месте, включает программу – она синхронизируется с сервером и отображает проект в масштабе 1:1 [5].

Также стоит сказать о современных проблемах технологии ДР, это: недоработка стандартов, низкая производительность аппаратных средств, низкая точность позиционирования, недостаточное распространение в силу новой технологии. Все эти проблемы носят инженерный характер и решаемы по мере развития технологии и научно-технического прогресса в целом. По мере устранения вышеперечисленных недостатков и совершенствования всех отраслей компьютерных технологий, ДР как продукт их синтеза достигнет всеобщего проникновения во все области человеческой деятельности.

В течение последующих 5–10 лет эта технология произведет очередной научно-технический переворот, сопоставимый с вездесущим распространением сети Интернет. Безусловно, всем известно, в каком темпе развивается Интернет, прошедший путь от двух соединенных проводом «калькуляторов» до настоящей всемирной паутины, которая начинает потихоньку выходить в космическое пространство. Не остается сомнений в том, что со временем дополненная реальность войдет в нашу повседневную жизнь так же как некогда смартфон или планшет, изменив ее раз и навсегда.

Литература

1. «Дополненная реальность – это просто» [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/144280/> (дата обращения: 06.12.2017).
2. «Принципы «дополненной реальности» и интерактивная реконструкция в строительстве», [Электронный ресурс]. – Режим доступа: <http://sias.ru/magazine/vypusk-6-2013/sotsialnaya-filosofiya-i-sotsiologiya/848.html/> (дата обращения: 06.12.2017).
3. «Рынок дополненной реальности» [Электронный ресурс]. – Режим доступа: <http://www.pvsm.ru/obrabotka-izobrazhenij/9013/> (дата обращения: 06.12.2017).
4. Cook J. et al. Real-time photo-realistic augmented reality for interior design // ACM SIGGRAPH 2003 Sketches & Applications. – 2016. – P. 1-1.
5. Pisman W. et al. Augmented reality with large 3D models on a PDA: implementation, performance and use experiences // Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry. – 2017. – P. 344–351.



Мустафина Ляйсан Ирековна

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4208

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: amix93@mail.ru

УДК 004

ИСПОЛЬЗОВАНИЕ БИБЛИОТЕК ДЛЯ ВЕБ-АНИМАЦИИ

Мустафина Л.И.

Научный руководитель – д.педагог.н., профессор Готская И.Б.

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

Анимация является хорошим инструментом, позволяющим повысить наглядность веб-продуктов, в том числе и их преимуществ. Популярность использования анимации привела к росту количества библиотек, ускоряющих и упрощающих процесс ее интегрирования. В работе рассмотрены наиболее популярные библиотеки и их возможности.

Ключевые слова: анимация, веб-дизайн, CSS, JavaScript, библиотека.

Анимация делает интерфейс удобнее и понятнее для обычного пользователя, позволяет более наглядно рассказать о продукте, в том числе продемонстрировать его преимущества. Анимацию можно использовать и в образовательных целях, повышая уровень наглядности информации.

Способы анимации отдельных элементов различного уровня входят в практику большинства разработчиков. Всплывающие баннеры, движущиеся логотипы, прокрутки на странице, раскрывающиеся формы стали привычными для пользователей. Популярность использования анимации привела к росту числа библиотек, ускоряющих и облегчающих процесс ее интегрирования. С ростом числа библиотек анимация значительно усложнилась. Теперь разработчики могут использовать ее, чтобы сделать свой продукт более оригинальным и запоминающимся.

Использование библиотек значительно облегчает и ускоряет процесс разработки. Однако при выборе определенной библиотеки разработчик должен точно осознавать стоящую перед ним задачу. Принимая во внимания недостатки использования той или иной библиотеки, стоит понимать, что неправильный выбор может значительно ухудшить качество итогового веб-продукта.

На данном этапе все библиотеки для веб-анимации можно разделить на два типа: библиотеки на основе CSS и библиотеки на основе JS. Они решают разные задачи, а значит, при выборе одной из них разработчик ориентируется на определенные критерии. Рассмотрим наиболее популярные на сегодняшний день библиотеки.

Animate.css – это простая в использовании библиотека охватывает в себе 77 видов анимации, сгруппированных по типу эффекта, включает в себя «вход» и «выход» из анимирования элемента, а также простой конструктор, который дает возможность включить в файл только нужные эффекты. Animate.css разработан Дэном Эденом, дизайнером и студентом из Манчестера (Великобритания) и находится в свободном доступе. Библиотеку уже используют сайты Foursquare, EA, Disney и т.д. [1].

Hover.css – это богатое собрание из 48 CSS hover-эффектов, разработанное Яном Лунном. Важным преимуществом этой библиотеки является то, что в архиве с библиотекой

присутствует Sass-версия. В связи с растущей популярностью CSS-препроцессоров в последнее время это стало очень удобным для большинства веб-разработчиков, использующих их. Библиотека подойдет для создания эффектов, которые будут срабатывать при наведении курсора мыши на элемент [2].

Velocity.js – очень мощная и хорошо оптимизированная библиотека с множеством функций. Среди них цветовая анимация, циклические преобразования, скручивание, прокрутки сцен и анимация SVG-элементов. Velocity.js полностью совместима с jQuery – достаточно добавить ее на страницу и заменить все вызовы \$.animate() на \$.velocity() [3].

D3.js – это набор инструментов для визуализации данных, что является отличительной особенностью этой библиотеки – D3.js включает несколько десятков небольших модулей, каждый из которых решает свою задачу. Кроме модулей для построения различных фигур, внутри D3.js есть модули для работы с элементами на странице (простой аналог jQuery), загрузкой данных (аналог fetch/\$.AJAX, заточенный под форматы CSV, JSON, XML и другие), форматированием и масштабированием данных, математическими функциями и другим. Базовая анимация в D3.js реализуется очень просто, дополнительно имеется богатый выбор методов для создания сложной и оригинальной анимации [4].

Greensock – одна из самых популярных библиотек для создания анимации, что обусловлено следующими преимуществами при использовании: удобный функционал, хороший уровень FPS, анимация любого CSS-свойства, отсутствие (или изменение) понятия ключевых кадров и т.д. [5].

С увеличением популярности библиотек для веб-анимации, простота создания динамических элементов пользовательского интерфейса стала играть важную роль в веб-разработке. Однако анимация должна оставаться не основным инструментом веб-дизайна, отсутствие анимации не должно сказываться на работоспособности интерфейса или считываемости контента сайта.

Таким образом, использование библиотек для веб-анимации меняет наше представление о современном сайте, открывая новые возможности для улучшения функциональности интерфейса. Изучение тенденций развития веб-анимации является важным вопросом в становление как веб-разработчиков, так и веб-дизайнеров.

Литература

1. Простая CSS3 анимация с Animate.css [Электронный ресурс]. – Режим доступа: <http://htmlbook.ru/blog/prostaya-css3-animaciya-s-animatecss> (дата обращения: 10.11.2017).
2. 5 библиотек для создания ярких CSS эффектов [Электронный ресурс]. – Режим доступа: <http://mattweb.ru/item/104-5-bibliotek-dlya-sozdaniya-yarkih-css-effektov> (дата обращения: 10.11.2017).
3. Velocity.js [Электронный ресурс]. – Режим доступа: <http://velocityjs.org/> (дата обращения: 11.11.2017).
4. Просто о D3.js [Электронный ресурс]. – Режим доступа: <https://habr.com/post/342106/> (дата обращения: 11.11.2017).
5. Greensock: анимация на JavaScript [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/144466/> (дата обращения: 11.11.2017).



Неудачин Василий Евгеньевич

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4108

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: neudachin1995@mail.ru

УДК 555.32

**АНАЛИЗ ИСТОЧНИКОВ ПО ТЕМЕ: «ИССЛЕДОВАНИЕ И СРАВНИТЕЛЬНЫЙ
АНАЛИЗ МЕТОДОВ ГЕНЕРАЦИИ ЭЛЕКТРОННЫХ ВАЛЮТ»**

Неудачин В.Е.

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 исследованы различные виды электронных валют и методы их генерации, а также отмечены плюсы и минусы каждого метода.

Ключевые слова: электронные деньги, криптовалюта, гибридный майнинг.

Человечество не стоит на месте, с развитием технологий появился универсальный метод оплаты, не требующий физического воплощения [1–5].

Электронные деньги – это серьезный катализатор экономического роста с момента их появления. Они представляют собой электронный эквивалент денежных средств, хранящихся на электронных носителях.

Из общего списка электронных валют можно выделить систему WebMoney, Яндекс-деньги, Qiwi, PayPal, как наиболее популярные и распространенные.

Наиболее распространенный и перспективный вид электронной валюты, на данный момент – криптовалюта, это вид цифровой валюты, защищенной от подделки, которую можно хранить в электронных кошельках, а также переводить из одного кошелька в другой.

Главными преимуществами криптовалюты являются: пиринговая архитектура (P2P), т.е. децентрализованная компьютерная сеть с абсолютным равноправием участников. Особые принципы и методы шифрования, используемые при эмиссии и учете криптовалюты. Функционирование криптовалюты определяется только правильностью алгоритма, заложенного в программу. Существует множество видов криптовалют: Bitcoin, NMC, Litecoin, FTC.

Криптовалюту можно генерировать (майнить), рассмотрим преимущество и недостатки наиболее популярных видов майнинга.

Proof of Work (PoW) – майнеры соревнуются между собой вычислительными мощностями. Чем больше мощностей у тебя задействовано, тем больше шанс добыть блок и получить с него комиссию.

Достоинства:

- надежность. Исключается злонамеренное влияние на сеть;
- требуются капитальные вложения. Виртуальные активы привязаны к реальным.

Недостатки:

- непроизводительные траты. Происходит «гонка» майнеров, вследствие чего сжигается энергия в больших масштабах;
- централизация. Крупные майнеры вытесняют мелких, и система становится уязвимой.

Proof of Stake (Pos) – работы ограничены реальной потребностью сети. Каждому ноду определена фиксированная ставка (stake).

Достоинства:

- токены работают на вас, но не теряются в процессе (можно сравнить с банковскими вкладами);
- токены исключаются из спекулятивного оборота («замораживаются» для фиксации одного нода).

Недостатки:

- низкая доступность по цене;
- существует теория о пониженной надежности, из-за централизации, однако, на данный момент, ни одна сеть PoS не была взломана.

Delegated Proof of Stake (DPoS) – делегированное доказательство доли, это усиленный PoS.

Достоинства:

- невероятная скорость сети;
- демократия. Все пользователи вовлечены в процесс управления сетью;
- хорошая управляемость, наличие механизмов принятия решений о будущем сети.

Недостатки:

- большая централизация. Делегатов ограниченное количество, но их легко заменить;
- спорный вопрос с надежностью, но, на данный момент эта сеть также не была взломана.

Гибридный майнинг – для того, чтобы достичь баланса между плюсами и минусами различных видов майнинга, часто применяется их объединение по самым разным схемам. Сейчас наиболее распространен гибридный PoW/PoS майнинг. Судя по всему он эффективнее чистого PoS и достаточно надежен, однако DPoS дает больше возможностей (но остается открытым вопрос рисков).

Литература

1. Vigna P., Casey M.J. The Age of Cryptocurrency How Bitcoin and the Blockchain Are Challenging the Global Economic Order. – St. Martin's Press, 2015. – 357 p.
2. Narayanan A., Bonneau J., Felten E., Miller A., Goldfeder S. Bitcoin and Cryptocurrency Technologies. – Princeton University Press, 2016. – 308 p.
3. Andreas M. Antonopoulos The internet of money. – CreateSpace Independent Publishing Platform, 2016. – 152 p.
4. Горюков Е.В. Электронные деньги: анализ практики использования и прогноз развития. – Иваново, 2004. – 162 с.
5. Красова Е.В. Иностранная рабочая сила в мировой и региональной экономике: современные аспекты государственного регулирования (монография). – Владивосток: Изд-во ВГУЭС, 2012. – 212 с.



Никитинская Вера Михайловна

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: nikitinskaya.v@gmail.com



Сокуренок Юрий Андреевич

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», к.т.н., доцент

e-mail: kpd@limtu.ru



Государев Илья Борисович

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: goss@corp.ifmo.ru

УДК 004.04

ОБЗОР СОВРЕМЕННЫХ ПРАКТИК СОЗДАНИЯ СТАТИЧЕСКИХ ВЕБ-САЙТОВ

Никитинская В.М., Сокуренок Ю.А., Государев И.Б.

Научный руководитель – к.т.н., доцент Сокуренок Ю.А.

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

В работе проведено сравнение характеристик статических и динамических сайтов. Выделены преимущества и недостатки статических веб-сайтов. Приведена классификация инструментов разработки и выполнен обзор стека JAM – современной практики создания статических сайтов.

Ключевые слова: статический веб-сайт, инструменты разработки сайтов, JAM-stack.

Первые веб-сайты были полностью статическими. Основной характеристикой статического сайта является размещение на сервере полностью готовых отдельных HTML-страниц, а также файлов, содержащих CSS-стили и JavaScript-код. При поступлении запроса сервер отдает клиенту готовые файлы. Соответственно, вся нагрузка по генерации страниц сайта смещается с момента получения запроса пользователя (как в динамических сайтах) на момент генерации сайта, которая производится заранее [1].

Растущая потребность в интерактивности веб-страниц привела к появлению динамических веб-сайтов. Основным отличием динамического сайта от статического становится момент генерации страницы: здесь она происходит непосредственно после получения запроса от пользователя и обработки его работающей на серверной стороне программой. В зависимости от содержания запроса пользователя серверное программное обеспечение выполняет различные инструкции, делает запросы к базе данных, формирует

содержимое страницы и затем отправляет страницу пользователю [1]. Данный процесс происходит для каждого пользователя и при каждом запросе.

Динамические сайты позволяют разрабатывать веб-приложения с богатым функционалом, однако обладают рядом недостатков, в основном в скорости и безопасности. В свою очередь, статические веб-сайты демонстрируют очень высокую скорость загрузки страницы и при этом надежны при непрогнозируемых пиковых нагрузках, что критически важно в современных условиях.

Многие функции, раньше требовавшие программирования на стороне сервера, теперь могут быть перемещены на клиентскую часть. Исходя из этого, такие динамические элементы, как комментарии, интеграция с соцсетями, поиск, чат в реальном времени могут быть подключены и к полностью статическому сайту в виде сервисов, работающих по HTTP-протоколу через API [2].

В российской литературе мало обращается внимание на однозначные преимущества статических сайтов, недостаточно описаны примеры работы с ними и не выделены ситуации, где использование статических сайтов не только оправдано, но и предпочтительно; основное внимание по-прежнему уделяется работе с динамическими веб-сайтами. При этом уже разработано большое разнообразие многофункциональных инструментов для создания и работы со статическими сайтами, хотя нет единых рекомендаций по целесообразности выбора тех или иных средств.

До сих пор встречается мнение о том, что статические сайты имеют крайне узкую область применения, сложны в разработке и не имеют никаких интерактивных возможностей [3]. Однако уже нельзя говорить о слишком узкой области применения данной технологии: статический сайт может как обладать отдельными интерактивными элементами, так и представлять собой полностью интерактивный инструмент, например, игру, калькулятор; функциональность ограничена лишь возможностями клиентского JavaScript [4].

В литературе приводят обширный список типов веб-сайтов, разработка которых в виде статического сайта может быть удобной и предпочтительной. Отмечается, что общей характеристикой таких ресурсов является ориентация на предоставление контента пользователю [2], а также относительно нечастое обновление содержимого, до 1–2 раз в день [4].

Статические сайты обладают рядом существенных преимуществ по сравнению с динамическими. Так, они более безопасны, поскольку отсутствует возможность проведения атак на базу данных (SQL-инъекции), внедрения пользовательского содержимого (XSS-атаки) и других [3]. Они практически не предъявляют особых требований к серверному окружению, нет необходимости устанавливать специфичный набор инструментов (например, MySQL и PHP определенной версии) [4]. Скорость отображения высока, поскольку после запроса пользователя отдача происходит сразу, без обращения к логике приложения и данным. По этой же причине при непредвиденных нагрузках приложение остается устойчивым, а при необходимости легко масштабируется простым копированием на дополнительные серверы готовых файлов JavaScript, HTML и CSS, полученных, например, в результате работы генератора статических сайтов. Есть возможность организовать размещение веб-сайта на CDN (content delivery network), что также дает преимущества в скорости загрузки и бесперебойной доступности сайта пользователям [5].

Один из способов разработки статических сайтов, JAM-стек (JavaScript, API, Markup) – это современная веб-архитектура, основанная на работе клиентского JavaScript, переиспользуемых API и языков разметки. Акцент делается на максимальную абстракцию серверной части веб-сайта от клиентской, т.е. отделение фронтенда от бэкенда. Серверная часть перестает существовать в привычном понимании и предоставляется через API от сторонних сервисов. Еще одна характеристика JAM-приложений – это модульность: каждый проект собирается индивидуально и только из необходимых компонентов, а при расширении функциональности другие компоненты легко подключаются к системе. Следующие признаки

характерны для JAM-сайта и считаются лучшими практиками для разработки статического сайта [5]:

- размещение всего сайта на CDN;
- модульная архитектура;
- размещение изменений проекта на сервере небольшими частями (атомарное);
- версионирование всего проекта с помощью системы Git;
- автоматизация сборки проекта.

В экосистеме JAM-стека уже создано большое количество разнообразных инструментов. Их можно разделить на следующие категории [2]:

- управление контентом и системы управления контентом (CMS, Content Management Systems);
- инструменты для размещения (хостинга);
- генераторы статических сайтов;
- темы визуального оформления;
- инструменты для взаимодействия с пользователями;
- прочие вспомогательные инструменты.

Например, для статических сайтов существуют специальные CMS, которые делают удобным редактирование контента сайта для неспециалистов. Другие инструменты позволяют добавить на статический сайт разнообразные интерактивные функции, например, обработку форм, аутентификацию пользователей, разделы комментариев. При разработке можно использовать инструменты непрерывной интеграции, проводить тестовые сборки перед итоговым размещением [2].

Статические веб-сайты до недавнего времени считались базовой технологией с крайне ограниченными возможностями и областью применения. Однако развитие клиентского программирования на языке JavaScript и различных программных средств и сервисов делают их привлекательным современным инструментом. При этом отмечается сложность и неопределенность, присущие ситуации выбора необходимых инструментов для конкретных проектов, а также недостаточная распространенность технологии. Кроме того, есть определенные ограничения в функциональности статических сайтов [5]. В российских источниках до сих пор мало информации о современном состоянии в разработке статических веб-сайтов, в основном рассматриваются вопросы создания динамических сайтов. Исходя из всего вышесказанного, представляется необходимым проводить дальнейшие исследования по данной теме.

Литература

1. Voucas E. An Introduction to Static Site Generators [Электронный ресурс]. – Режим доступа: <https://davidwalsh.name/introduction-static-site-generators> (дата обращения: 20.01.2018).
2. Camden R., Rinaldi B. Working with Static Sites. – O'Reilly Media, Inc., 2016. – 195 p.
3. Першина Т.А., Сгибнева С.С. Сравнительная характеристика статических и динамических сайтов // Теория и практика современной науки. – 2016. – № 8. – С. 320–322.
4. Rinaldi B. Static Site Generators. – O'Reilly Media, Inc., 2015. – 57 p.
5. JAMstack for Clients: Benefits, Static Site CMS, & Limitations [Электронный ресурс]. – Режим доступа: <https://snipcart.com/blog/jamstack-clients-static-site-cms> (дата обращения: 20.01.2018).

**Нифанина Анастасия Романовна**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4105Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: nansea@yandex.ru

УДК 004.92**ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ГРАФИКИ ДЛЯ ПРОДВИЖЕНИЯ ТОВАРОВ
И УСЛУГ****Нифанина А.Р.****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 исследованы основные задачи и возможности SMM-продвижения товаров и услуг, описаны способы эффективного взаимодействия с аудиторией, рассмотрены тенденции, преобладающие в процессе продвижения товаров и услуг, выделены площадки для осуществления SMM-продвижения в России.

Ключевые слова: интернет, социальные сети, SMM-продвижение, графика, графические технологии.

Особенностью и тенденцией развития графических технологий для продвижения товаров и услуг в наши дни стало их все большее и большее их в сети Интернет. За 2017 год по данным АКАР объем рекламы в Интернете практически сравнялся с телевизионной рекламой (+23% и 115,0–116,0 млрд руб. против 116,0–117,0 млрд руб. (+13%) соответственно) [1]. Более того, прослеживается тенденция, что подавляющее большинство компаний поддерживают свои сайты и интернет-магазины аккаунтами в социальных сетях.

Максимальное распространение социальных медиа обусловлено эффективностью этого канала коммуникации. Это отмечают владельцы как крупных и уже всем давно известных брендов, так и небольшие стартапы. Разработчики программного обеспечения (ПО) регулярно внедряют функции и возможности графики, способные удерживать внимание пользователей на еще большее время, чем ранее. За счет этого появляются новые инструменты для продвижения в социальных сетях, и возникает необходимость в специализированных кадрах.

С течением времени для работы в этом сегменте появилась отдельная профессия SMM (Social Media Marketing)-менеджера, основной функцией которого и является организация продвижения товаров, услуг и брендов в социальных сетях.

Наиболее популярными на российском рынке социальными сетями являются: ВКонтакте, Facebook, Instagram, Одноклассники и Twitter. Более 80% россиян ежедневно используют социальные сервисы, проводя в них в среднем 2,5 ч, а суммарная аудитория активных пользователей площадок в российском сегменте составляет более 20 млн. К сожалению, присутствие компании во всех социальных сетях и большое количество подписчиков не гарантируют эффективность вашего проекта. В связи с этим появляется вопрос: какой контент наиболее продуктивен для выполнения бизнес-задач в различных социальных сетях?

Самыми распространенными и используемыми графическими объектами в социальных сетях являются обработанные фотоматериалы, иллюстрации, а также GIF (Graphics Interchange Format)-анимации и видеоматериалы разного размера и сложности. Следовательно, использование графических технологий перестает быть прерогативой лишь графических-дизайнеров и моушн-дизайнеров.

Для выяснения предполагаемых компетенций в области компьютерной графики, необходимых для специалистов по интернет-продвижению, было проведено несколько опросов.

Один из них – среди пользователей сети. Он позволил выяснить, что кажется конечному потребителю привлекательным и какие социальные сети наиболее популярны.

Как показал опрос самая посещаемая сеть – Instagram, на втором месте – ВКонтакте, на третьем – Telegram и на четвертом – Facebook.

По мнению респондентов, самой эффективной для продвижения социальной площадкой является Instagram (60% голосов). Это и не удивительно. Разработчики Instagram регулярно добавляют различные функции и совершенствуют уже существующие возможности, что позволяет привлекать внимание конечных потребителей к товару или услуге, повышая в конечном итоге эффективность использования социальных медиа для увеличения прибыли и расширения рынков сбыта.

Анализ ответов на вопрос «Что больше всего привлекает в различных аккаунтах в социальных сетях?» показал, что респондентов больше всего привлекает наличие собственного стиля в оформлении аккаунтов (32% голосов), красивые фото (28%). По их же мнению, видеоконтент тоже необходим в аккаунтах.

Анализ активности и откликов на публикации свидетельствует, что подписчики аккаунтов в социальных сетях лучше всего реагируют на публикации с привлекательными фотографиями людей и указанием цифровых или статистических данных (цена, например, или заголовок «7 фактов о...»). Пользователи любят видеть результат, поэтому рубрики «До/После» чаще всего набирают максимальное количество откликов. Они могут быть представлены как в виде коллажа, так и как просто в виде видеоролика, который демонстрирует трансформацию. Статистика показывает, что большинство людей просматривают социальные сети с мобильных телефонов, поэтому длинные и сложные ролики не получают ожидаемого отклика. Особое внимание пользователей привлекают лишь вирусный (тот, которым начинают делиться) и шок-контент (провокационный) или анимационные или видеоролики без текстов, но с популярной музыкой. Полноценный же видеоматериал стоит оставить для социальных сетей, просмотр которых возможен с персональных компьютеров.

Еще один опрос был проведен среди SMM-специалистов. Его результатом стало понимание, какие технологии наиболее востребованы для удовлетворения потребностей клиента, какое ПО используется чаще всего для разработки графического контента и кто его разрабатывает для публикации в социальных сетях.

Опрос показал, что 75% SMM-специалистов готовят графический контент самостоятельно, из них более половины (53%) испытывают трудности при его подготовке.

Основные обязанности SMM-специалиста: фото и видеосъемка, обработка фотоматериалов, создание баннеров, GIF-анимаций и простейших видео, работа с готовым оформлением аккаунта (шаблонами).

Большинство респондентов предпочитают компьютерные графические редакторы мобильным. С видео редакторами складывается обратная ситуация, скорее всего потому, что они достаточно сложные в использовании и требуют специальных знаний.

80% респондентов были бы рады улучшить свои навыки работы в различных редакторах.

Следовательно, насущной потребностью является повышение дизайнерской компетентности специалистов, которые в большинстве своем ориентированы на работы с текстовым контентом, т.е. существует проблема работы с графическими технологиями, без которых работа с социальными сетями для продвижения товаров и услуг становится практически невозможной.

Одним из путей решения этой проблемы будет создание специализированного курса. Это необходимо, потому что профессия SMM-щика предполагает оперативное реагирование на изменения деловой ситуации, умение и возможность разрешать форс-мажорные обстоятельства в любых условиях. К тому же, большинство менеджеров по рекламе и связям с общественностью полностью самостоятельно ведут аккаунты своих компаний в различных социальных сетях, и в целях оптимизации расходов обращение к сторонним дизайнерам не предполагается. Другой целевой аудиторией для создаваемого курса могут быть предприниматели, которые занимаются продвижением своего бизнеса в Интернете без привлечения специалистов и хотят делать это грамотно и красиво.

Программа курса будет предполагать, обучение SMM-специалистов работе с Adobe Photoshop (обработка фотографий, создание рекламных баннеров, простейшая анимация), использовании программы линейного монтажа Adobe Premier Pro и базовым навыкам созданию анимаций Adobe After Effects. Также дополнительно можно будет рассмотреть бесплатное мобильное ПО, которое поможет в решении оперативных задач.

Курс возможен в форме очного обучения, где различные специалисты по той или иной теме будут проводить лекции и практические занятия. Но актуальнее будет проводить курс дистанционно, где его участники смогут в любое время познакомиться с записанными лекциями и изучить подготовленные конспекты и к определенным срокам выполнять практические задания, получая обратную связь с подробным разбором занятий от ведущих курса.

Подводя итог вышесказанному, необходимо отметить, что стремительно развивающиеся графические технологии и все более возрастающая роль социальных медиа в жизни каждого человека предполагают дальнейшее продуктивное использование социальных сетей для продвижения товаров и услуг в сети Интернет. Следовательно, необходима потребность в высококвалифицированных специалистах, владеющих не только умением качественно формулировать преимущества той или иной услуги или товара, создавать текстовый новостной контент, но и легко решающих любые графические задачи [2–5].

Литература

1. Медиапрофи [Электронный ресурс]. – Режим доступа: <http://mediaprofi.org> (дата обращения: 15.01.2018).
2. Аналитика Рунета [Электронный ресурс]. – Режим доступа: <https://therunet.com/analytics> (дата обращения: 15.01.2018).
3. Нетология – университет интернет-профессий [Электронный ресурс]. – Режим доступа: <https://netology.ru/blog/vybor-smm> (дата обращения: 25.01.2018).
4. Стелзнер М. Контент-маркетинг. Новые методы привлечения клиентов в эпоху интернета. – М.: Манн, Иванов и Фербер, 2012. – 70 с.
5. Халилов Д. Маркетинг в социальных сетях. – М.: Манн, Иванов и Фербер, 2014. – 65 с.



Павлихин Сергей Валерьевич

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: sergey.pavlikhin@inbox.ru

УДК 004

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕХАНИЗМОВ ОБМЕНА ДАННЫМИ МЕЖДУ
ВЕБ-КЛИЕНТОМ И ВЕБ-СЕРВЕРОМ**

Павлихин С.В.

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 рассмотрены механизмы обмена данными между веб-клиентом и веб-сервером, выявлены их слабые и сильные стороны.

Ключевые слова: веб-приложение, Comet, Polling, Long polling, WebSocket, Server Sent-Events.

С развитием Интернета росли и возможности веб-браузеров, что позволило разработчикам создавать многофункциональные веб-приложения, которые по своей сложности и возможностям почти не уступают десктопным приложениям [1].

Все веб-приложения построены на клиент-серверной архитектуре, где в качестве клиента выступает браузер, а в качестве сервера – веб-сервер. Клиент делает запрос к какому-нибудь ресурсу, а веб-сервер обрабатывает этот запрос и возвращает ответ. На таком принципе основываются все механизмы обмена данными между веб-клиентом и веб-сервером, однако между ними есть различия в способах, частоте, форматах запроса и ответа.

Эти различия можно наглядно представить в виде сравнительного анализа, на основе критериев: объем данных, актуальность данных, поддержка браузерами, двусторонняя связь. В рамках данной работы были рассмотрены следующие механизмы обмена данными: запрос-ответ, опрос, длительный опрос, Server-Sent Event, WebSocket.

Механизм «запрос-ответ» самый примитивный и является родителем для все остальных. Веб-клиент делает запрос только тогда, когда пользователь захочет, чтобы данные на клиенте или сервере обновились. Например, нажатие по кнопке, ввод значения в поле ввода и т.д. Объем передаваемых данных зависит от количества действий пользователя, но есть лишние данные, а именно заголовки, которые возможно поменяются лишь один раз. Актуальность данных также зависит от количества действий пользователя, но в автономном режиме она минимальна. Двусторонний обмен возможен и поддерживается без ограничений. Данная схема неприемлема для приложений, которые должны предоставлять информацию в реальном времени, которых на сегодняшний день большинство [2]. Это связано с тем, что при таком подходе у сервера нет возможности принудительно дать клиенту информацию, или заставить его попросить эту информацию. Для обхода такого ограничения используют другие механизмы, которые основаны на технологии AJAX и подхода Comet.

Одним из таких механизмов является «опрос» (polling). Этот способ самый простой и основан на постоянном обращении клиента к серверу за новой информацией в фоновом режиме. Веб-клиент отправляет AJAX-запросы с заданной временной периодичностью, тем самым опрашивает веб-сервер о наличии новых данных или событий. Веб-сервер, в свою

очередь, готовит ответ для каждого запроса и отправляет их обратно веб-клиенту. Если новых данных не появилось, то сервер все равно отвечает клиенту.

Очевидно, в данном подходе количество передаваемых данных зависит, собственно, не только от самих данных, но и от интервала между запросами. При достаточно маленьком интервале объем данных может превысить размер всей страницы за пару минут. Более того, очень много лишних запросов, которые посылаются клиентом на сервер, но там еще нет новых данных или событий. Актуальность данных зависит от интервала между запросами. Поддерживается всеми браузерами, где работает JavaScript. Обмен данными двусторонний.

Более совершенными транспортными технологиями являются технологии под общим термином Comet. Comet – это коллекция техник, которые при постоянном (или длительном) HTTP-соединении позволяют веб-серверу отправлять данные браузеру без дополнительного запроса с его стороны. Общая черта таких моделей состоит в том, что все они основаны на технологиях, непосредственно поддерживаемых браузером.

Более совершенным механизмом обмена данными является «длительный опрос» (long polling), который также работает на обычном HTTP. После полной загрузки веб-страницы и готовности начать обмен, клиентский код посылает запрос на веб-сервер, но тот не отвечает сразу и не разрывает соединение. Ответит сервер тогда, когда появятся новые данные и затем разорвет соединение (согласно протоколу HTTP). Веб-клиент, после получения ответа, сразу же отправляет следующий запрос, тем самым снова запускает процесс ожидания новых данных на веб-сервере.

Данный механизм решает недостатки предыдущего механизма, а именно: сокращает количество запросов, убирает запросы «впустую»; высокая актуальность данных, и сохраняет достоинства: поддержка все браузерами и двусторонний обмен, но передаваемые данные все же имеют лишнюю, не изменяемую от запроса к запросу информацию – служебные HTTP-заголовки.

При каждом подключении к серверу отправляется полный набор HTTP-заголовков, и с ростом числа запросов это становится настоящей проблемой. Стоит сказать, что связь между веб-клиентом и веб-сервером возможна и через обычное TCP-соединение, однако для этого придется использовать дополнительные модули для браузеров, такие как Adobe Flash, Silverlight и Java. Они позволяют веб-серверам принудительно отправлять клиентам оперативные данные. Но использование сторонних модулей влечет за собой большой риск, что на клиентской стороне вообще ничего не будет работать. Дело в том, что нет никаких гарантий, что в клиентском браузере они установлены, а также часто случаются конфликты этих модулей и с локальными брандмауэрами, и особо часто – в корпоративных сетях.

Спецификация HTML5 определила несколько новых технологий, которые решали все вышеуказанные недостатки.

Одна из таких технологий – Server-Sent Events (SSE). Данная технология предоставляет специальный объект EventSource, через который устанавливается соединение с сервером один раз. Это объект в случае разрыва связи делает пересоединение и генерирует специальные события при поступлении данных с сервера или при ошибках.

После инициализации экземпляра EventSource браузер отправит запрос на установление соединения с сервером. Чтобы соединение было успешным, сервер должен ответить со значением в «Content-Type», равным «text/event-stream», и не закрывать соединение. После этого сервер может отправлять данные клиенту при необходимости. Server-Sent Events позволяет отправлять идентификатор сообщения, что помогает восстановить потерянные события и данные в случае обрыва связи.

Очевидно, что данный механизм значительно уменьшает количество передаваемых по сети лишних данных, так как HTTP-заголовки передаются только при установлении соединения. Сообщения не требуют этих заголовков. Актуальность данных при таком подходе максимальная и поддерживает такой механизм во всех браузерах, которые

поддерживают HTML5. Однако передача данных возможно только от сервера к клиенту, что накладывает определенные ограничения в использовании.

Альтернативное решение, которое тоже пришло с HTML5 – это технология WebSocket. По своей сути, это протокол, предоставляющий двунаправленную полудуплексную связь, который работает с протокола TCP. Организация W3C занимается стандартизацией WebSocket API (программного интерфейса) для доступа к протоку из веб-браузера, а сам протокол определяется стандартом RFC 6455 [3]. Переключение с HTTP на WebSocket-протокол происходит после обмена специальными заголовками между веб-клиентом и веб-сервером. Если сервер в ответных заголовках подтверждает свою готовность использовать WebSocket, то дальше обмен сообщения идет согласно протоколу. Протокол WebSocket не имеет с HTTP ничего общего, потому для его использования требуется специальное программное обеспечение на сервере (в отличие от всех предыдущих механизмов).

После установки соединения сервер и клиент могут посылать друг другу сообщения, как только это необходимо. Таким образом, актуальность данных максимальная и в обе стороны. Формат передаваемых пакетов (фреймов) имеет компактную оптимальную структуру, и потому объем накладных расходов на передачу данных стремится к минимуму, что, в свою очередь, повышает производительность. Кроме того, фреймы делятся на два больших типа: фреймы с данными и управляющие фреймы, предназначенные для проверки связи (ping) и закрытия соединения. Помимо передачи текстовой информации, фреймы с данными могут содержать и бинарные данные.

WebSocket также поддерживает безопасные соединения. Поддерживается во всех браузерах, которые поддерживают HTML5.

Рассмотрев все механизмы обмена данными между веб-клиентом и веб-сервером, можно наглядно представить получившиеся результаты сравнительного анализа в виде таблицы.

Таблица. Результаты сравнительно анализа

Механизм	Объем данных	Актуальность данных	Двусторонняя связь	Поддержка браузерами
Запрос-ответ	Зависит от количества действий пользователя, плюс заголовки	Зависит от количества действий пользователя	Полностью поддерживается	Поддерживается всеми браузерами
polling	Зависит от интервала между запросами, плюс заголовки	Зависит от интервала между запросами	Полностью поддерживается	Поддерживается всеми браузерами
Long-pollig	Зависит от самих данных, плюс заголовки	Максимальная	Полностью поддерживается	Поддерживается всеми браузерами
Server-Sent Events	Зависит только от самих данных	Максимальная	Поддерживает связь только от сервера к клиенту	Поддерживается всеми браузерами, которые поддерживают HTML5
WebSocket	Зависит только от самих данных	Максимальная	Полностью поддерживается	Поддерживается всеми браузерами, которые поддерживают HTML5

В результате работы был проведен анализ существующих механизмов обмена данными. Были обозначены сильные и слабые стороны каждого механизма, что может помочь разработчикам в выборе инструментов для построения веб-приложений.

Литература

1. Рогазинский А.А., Сидоров А.А. Использование подхода Redux при проектировании архитектуры веб-приложения [Электронный ресурс]. – Режим доступа: [https://sibac.info/archive/technic/4\(51\).pdf](https://sibac.info/archive/technic/4(51).pdf), своб.
2. Zhao S.M. et al. A Real-Time Web Application Solution Based on Node.js and WebSocket // Advanced Materials Research. – 2013. – V. 816–817. – P. 1111–1115.
3. Стандарт RFC 6455 [Электронный ресурс]. – Режим доступа: <https://tools.ietf.org/html/rfc6455>, своб.



Парфенова Ольга Владимировна

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: Olya.Parfyonova74@yandex.ru



Готская Ирина Борисовна

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, д.педагог.н., профессор

e-mail: iringot@mail.ru

УДК 004.9

АНАЛИЗ ПОПУЛЯРНОСТИ CMS-СИСТЕМ В 2017 ГОДУ

Парфенова О.В.

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 приведен анализ популярности CMS систем по данным за 2017 год путем анализа запросов в поисковой системе Google. Данные рассматриваются отдельно по всему миру и по России, выделены наиболее популярные CMS-системы, проведен анализ причин их популярности.

Ключевые слова: CMS-системы, популярность CMS, веб-разработка, создание сайтов, использование CMS, система управления контентом.

В настоящее время интернет-технологии развиваются крайне стремительно. Выбор технологий разработки достаточно широк, и все эти технологии отличаются друг от друга по безопасности, скорости и сложности разработки, скорости работы и легкости сопровождения.

CMS (Content management system)-система – готовая система управления, которая устанавливается на хостинг и включает дополнения для расширения функциональности [1]. Системы бывают как платные, так и бесплатные. Большой проект можно построить даже на бесплатной CMS без участия программистов, однако требуется опыт веб-разработки, чтобы правильно установить, настроить и обеспечить безопасность CMS.

Самые популярные CMS-системы [1]:

- WordPress – изначально движок для блога, сейчас на ней делаются практически любые сайты, включая интернет-магазины. Одна из самых популярных CMS в мире;
- Joomla – CMS общего назначения. Обычно на ней делают очень маленькие сайты, и обычно дешевле всех других вариантов. Проста в установке и управлении, достаточно надежна;
- Drupal – довольно мощная система общего назначения. Интуитивно понятный интерфейс позволяет строить сайты без знания верстки и HTML. Подходит для создания сайтов и веб-приложений различного назначения и скорости [2];

- ModX – проста в освоении, профессиональная CMS, подходит для создания практически любых сайтов [3];
- 1С-Битрикс – коммерческая CMS-система общего назначения с очень широкими возможностями. Подходит для создания больших и сложных сайтов [4];
- UMI.CMS – коммерческая система, содержит в себе все новейшие разработки и предоставляет широкие возможности для создания любого типа сайта. Несложна в освоении и имеет хорошую скорость работы [5].

Ниже приведена статистика запросов Google на основании сервиса trends.google.ru за 2017 год. Рассматривается сравнение количества запросов среди бесплатных CMS, среди платных CMS и сравнивается степень популярности платных и бесплатных CMS путем сравнения количества запросов наиболее популярной платной и бесплатной системы. Приведена статистика по всему миру и отдельно по России.

Рис. 1 отражает популярность бесплатных CMS в России. Из графика видно, что частота запросов по WordPress более чем в 2 раза превышает количество запросов по Joomla, и более чем в 4 раза превышает количество запросов по ModX и Drupal, из чего следует уверенное лидерство WordPress среди бесплатных CMS.

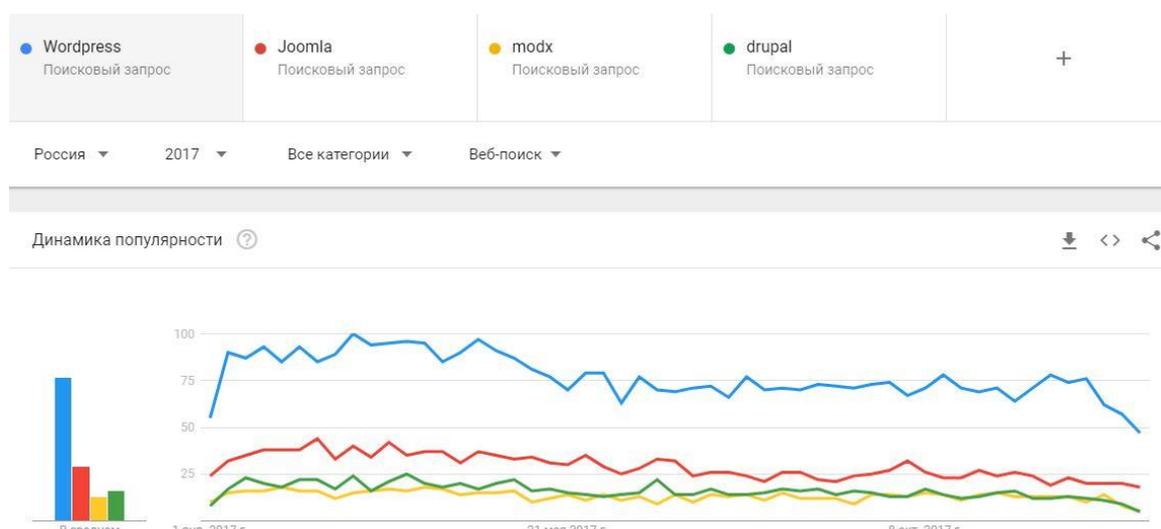


Рис. 1. Популярность бесплатных CMS в России

Рис. 2 иллюстрирует популярность бесплатных CMS по всему миру. Из графика видно, что во всем мире WordPress еще более популярна, чем в России.

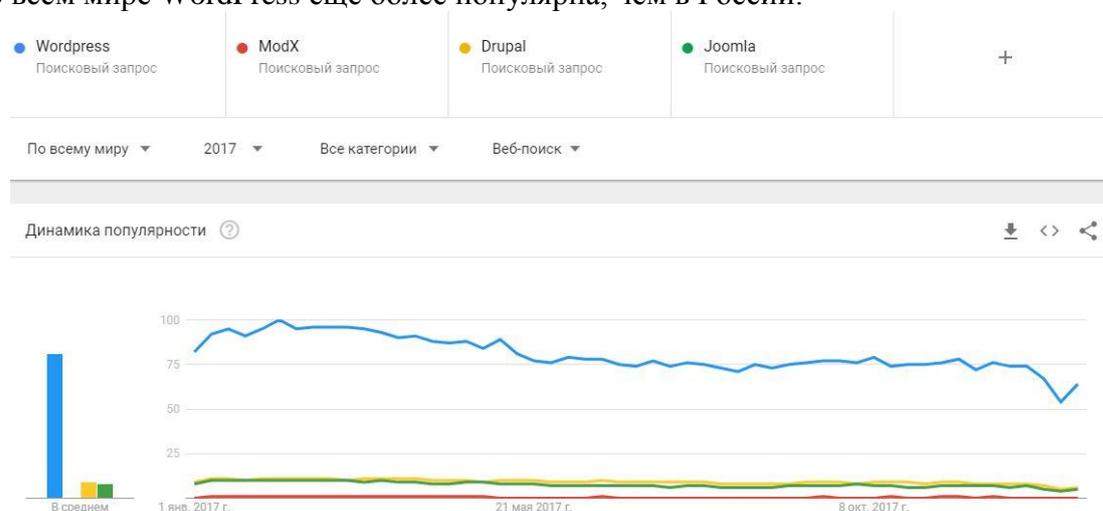


Рис. 2. Популярность бесплатных CMS в мире

На рис. 3 представлена сравнительная популярность 1С-Битрикс и UMI.CMS, а на рис. 4 отражена популярность этих же систем по миру. Очевидно, что разрыв еще более значителен.

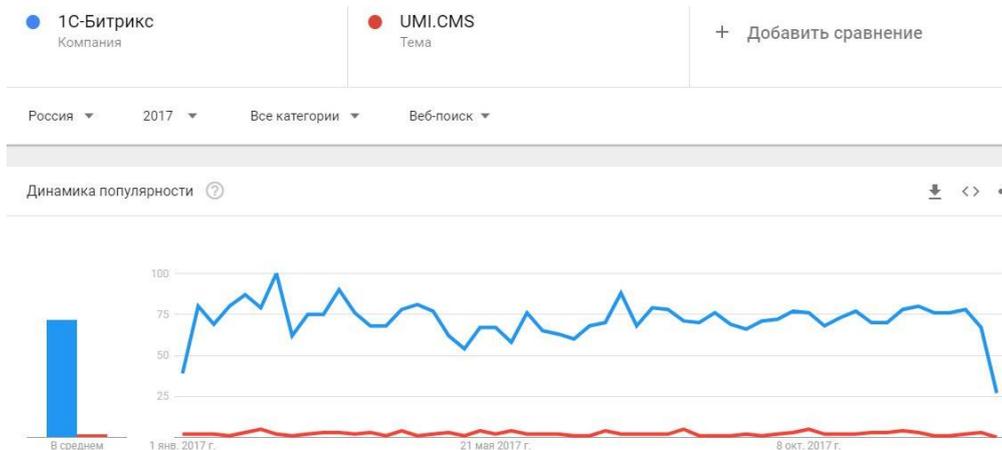


Рис. 3. Популярность коммерческих CMS в России

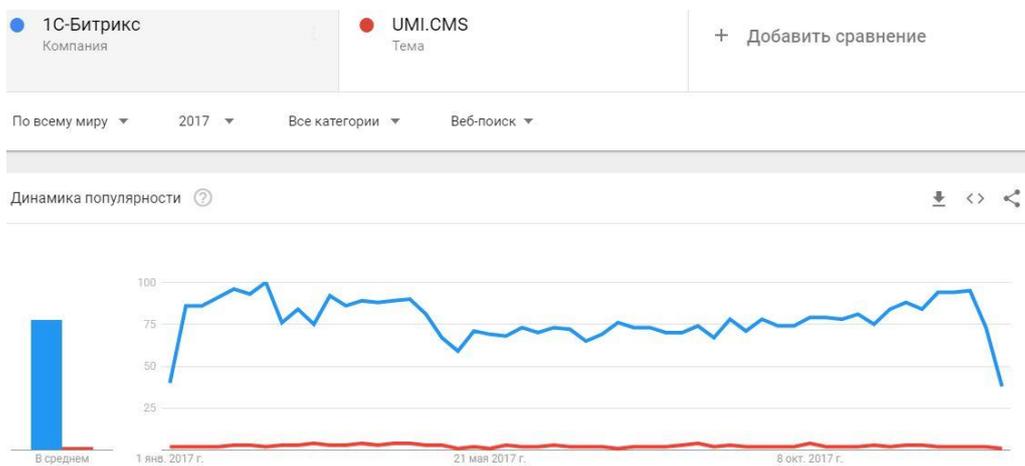


Рис. 4. Популярность коммерческих CMS в мире

На рис. 5 отражена степень популярности бесплатной CMS WordPress в сравнении с коммерческой 1С-Битрикс. Из графика видно, что популярность WordPress почти в 2 раза превышает популярность 1С-Битрикс. Рис. 6 иллюстрирует популярность WordPress и 1С-Битрикс по всему миру.

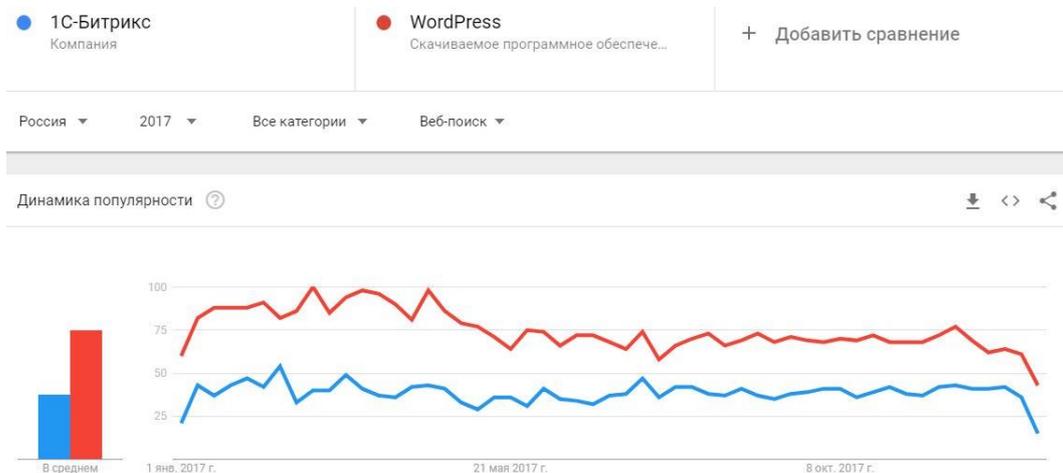


Рис. 5. Популярность коммерческих и бесплатных CMS в России

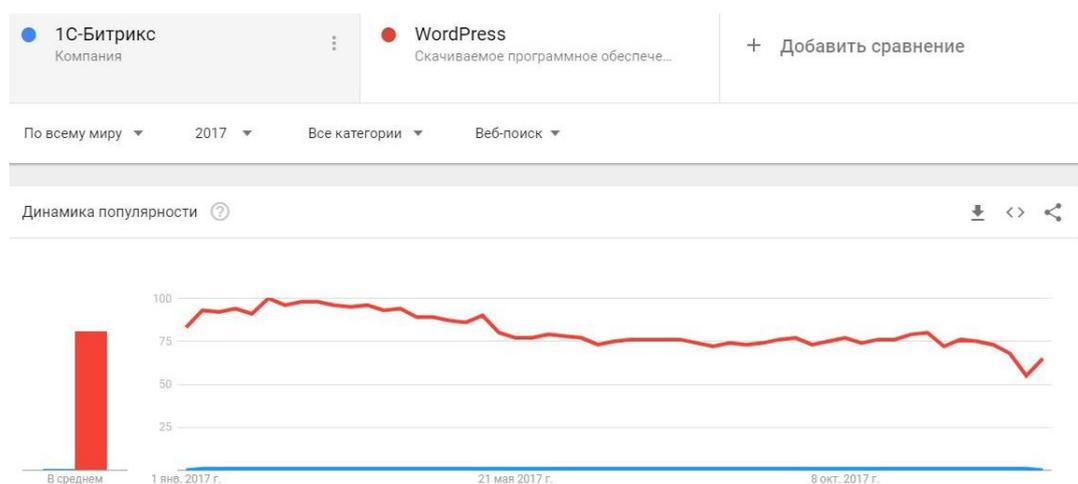
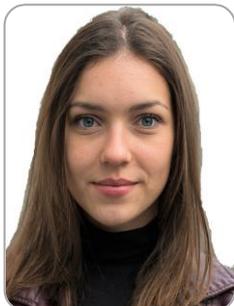


Рис. 6. Популярность коммерческих и бесплатных CMS в мире

На основании проведенного исследования был сделан вывод, что наиболее популярная CMS-система во всем мире – CMS WordPress, которая в большей степени популярна во всем мире, чем в России. В России в несколько раз чаще используется коммерческая система 1С-Битрикс, чем в мире, однако среди лидеров в разной степени популярности по России и по миру одни и те же CMS-системы [6].

Литература

1. Самостоятельное создание сайта – выбираем подходящий инструмент [Электронный ресурс]. – Режим доступа: <https://konstruktorysajtov.com/samostoyatelnoe-sozdanie-sajta> (дата обращения: 12.01.2018).
2. Горнаков С.Г. Осваиваем популярные системы управления сайтом (CMS). – М.: ДМК Пресс, 2009. – 336 с.
3. Обзор CMS ModX [Электронный ресурс]. – Режим доступа: <https://konstruktorysajtov.com/modx> (дата обращения: 28.01.2018).
4. Обзор CMS 1С-Битрикс [Электронный ресурс]. – Режим доступа: <https://konstruktorysajtov.com/cms/1c-bitrics> (дата обращения: 28.01.2018).
5. Сайт Конструкторы сайтов [Электронный ресурс]. – Режим доступа: <https://konstruktorysajtov.com/cms/umicms> (дата обращения: 28.01.2018).
6. Выбор технологий для большого и не очень большого веб-проекта [Электронный ресурс]. – Режим доступа: https://habrahabr.ru/company/SECL_GROUP/blog/315734/ (дата обращения: 12.01.2018).



Петрова Екатерина Юрьевна

Год рождения: 1996

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4108

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: petrovae9@yandex.ru



Государев Илья Борисович

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: goss@corp.ifmo.ru

УДК 004.04

**АНАЛИЗ ИСТОЧНИКОВ ПО ТЕМЕ: «ПРОБЛЕМА РАСШИРЕНИЯ
ФУНКЦИОНАЛЬНОСТИ КОРПОРАТИВНЫХ ПОРТАЛОВ»**

Петрова Е.Ю., 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 исследован вопрос применения различных методов увеличения стандартного функционала корпоративных порталов, на примере MS Sharepoint. Рассмотрены варианты применения технологий для расширения функциональности корпоративных порталов.

Ключевые слова: корпоративный портал, MS Sharepoint, PowerShell, функционал, JavaScript.

Рост необходимости информационного обеспечения на предприятиях повлек за собой появление программных продуктов для создания корпоративных порталов и электронного документооборота. Стандартные решения, основанные на платформах для создания корпоративных порталов, не полностью удовлетворяют потребностям организаций и требуют доработки. Разнообразие требований к корпоративным порталам повлекло за собой увеличение способов расширения стандартного функционала.

Расширение стандартного функционала корпоративного портала на базе одной из самых распространенных платформ MS Sharepoint [1] возможно с помощью различных технологий и языков программирования, самыми распространенными из них являются: HTML, CSS, PowerShell, JavaScript, SQL, SSRS.

Стандартный функционал корпоративных порталов на базе готовых платформ, в частности MS Sharepoint, обширен, но внедрение дополнительного функционала делает систему развитой и кастомизированной. Доработка стандартного портала происходит с помощью языков веб-программирования, так как сами порталы используются в браузерах.

Каждый язык программирования или технология используются для определенных задач. Например, HTML и CSS используются для изменения визуального оформления страницы, для изменения дизайна [2]. Важной частью доработки стандартного функционала является изменение дизайна, как пишут авторы книги [2] «основной целью дизайна Sharepoint является создание отличного пользовательского интерфейса для тех, кто посещает

сайт». Примерами использования HTML и CSS является: добавление подсказки в формы создания элементов в MS Sharepoint, задание элементам DOM определенного размера. Для изменения стандартного дизайна веб-страниц корпоративного портала, также используется XSLT – язык преобразований XML-документов, чаще всего применяется для визуального оформления веб-частей.

JavaScript является распространенным языком для увеличения функциональности корпоративных порталов и используется для решения разнообразных задач. JavaScript применяется для изменения дизайна веб-страниц, тем самым в значительной степени может заменить HTML, CSS. Также JavaScript используется для взятия данных из библиотек и списков корпоративного портала, из Active Directory MS Sharepoint. Полученную информацию JavaScript может автоматически вставлять в поля форм или применять для других целей. Данный язык используется для создания интерактивности на веб-страницах корпоративного портала [3]. Библиотека языка JavaScript – JQuery используется для динамических эффектов в интерфейсе, данную библиотеку можно подключить глобально на корпоративный портал и применять JQuery во всем программном коде. Также используется JSLink – это средство, которое управляет рендерингом полей, элементов и даже веб-фрагментов (Web Parts) через JavaScript-файл. Рендеринг на клиентской стороне относится к отображению данных на странице с применением технологий, работающих на клиенте, таких как JavaScript, HTML и CSS.

Следует обратить внимание на фреймворки, используемые для работы с Microsoft SharePoint, например, Sencha Ext JS. Sencha Ext JS может использоваться для создания задачника, консолидирующего задачи рабочих процессов портала. Sencha Ext JS позволяет создавать интегрированные компоненты MS Sharepoint с высокой производительностью, таким образом, Sencha Ext JS повышает интерактивность интерфейса.

PowerShell является основным средством администрирования корпоративных порталов [4]. С помощью интегрированной среды сценариев – Windows PowerShell ISE можно создавать, удалять, редактировать списки, библиотеки. Помимо этого, PowerShell имеет применение в обработчике событий MS Sharepoint – Item Receiver, List Receiver. Программный код из Receiver запускается при определенном действии над элементом, списком, библиотекой. Такими действиями могут быть: добавление, изменение удаление элемента, а элементами могут быть списки, элементы списка.

Для создания отчетности на базе корпоративного портала используются службы Reporting Services и SQL. С помощью языка структурированных запросов SQL берутся данные из баз данных MS Sharepoint [5].

Службы Reporting Services могут быть интегрированы с продуктами Sharepoint, что увеличивает функциональные возможности портала. Например, в качестве веб-частей на страницах корпоративного портала можно использовать отчеты, построенные на Reporting Services, чаще всего такими отчетами могут являться протоколы рабочих процессов или небольшие аналитические отчеты, так как от объема отчета зависит скорость загрузки страницы. Отчеты, созданные на основе Reporting Services, позволяют дополнить функционал Sharepoint табличными, графическими, интерактивными веб-частями.

Для реализации одной и той же задачи могут использоваться решения с разными языками программирования и с разными технологиями. Например, скрытие полей в форме создания элемента может быть реализовано как с помощью JavaScript, так и с помощью JQuery, HTML, CSS. Но существуют задачи, которые могут быть решены только с помощью одного языка программирования или технологии, например, администрирование корпоративного портала и обработку событий применяют только PowerShell.

Доработка стандартного функционала является неотъемлемой частью внедрения корпоративного портала в компанию. Совершенствование функционала позволяет значительно расширить стандартные возможности корпоративного портала, создать

уникальный продукт, повысить его ценность и эффективность с точки зрения требований бизнеса к используемым информационным технологиям.

Литература

1. Кирьянов Д.В. Разработка приложений Web 2.0 на Microsoft Sharepoint. – 2-е изд. – НОУ «Интуит», 2016. – 370 с.
2. Brandon A. Custom Sharepoint Solution with HTML and Javascript. – Apress, 2015. – 227 p.
3. Drisgli R. Sharepoint 2013 branding and user interface design. – Wrox, 2013. – 432 p.
4. Mann S., PowerShell for Sharepoint 2013 How-To. – Sams, 2013. – 203 p.
5. Turley P., Microsoft SQL Server Reporting Services Recipes: for Designing Expert Reports. – Wrox, 2010. – 319 p.

**Полетова Алина Игоревна**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4106Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: shock_harrison@yahoo.com

**Шуклин Дмитрий Анатольевич**

Год рождения: 1979

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: do@limtu.ru

УДК 004.92

**АНАЛИЗ СОВРЕМЕННЫХ ПОДХОДОВ К UX-ДИЗАЙНУ МОБИЛЬНЫХ
ПРИЛОЖЕНИЙ****Полетова А.И.****Научный руководитель – к.педагог.н., доцент Шуклин Д.А.**

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

В работе рассмотрены особенности современных подходов к оценке опыта взаимодействия и его повышению в процессе UX-проектирования мобильных приложений.

Ключевые слова: UX-дизайн, мобильные приложения, тестирование мобильных приложений.

Сегодня смартфоны по частоте использования многократно превосходят планшеты и домашние компьютеры. Установлено, что в России за 2017 год количество мобильных пользователей увеличилось на 15%, а тех, кто выходит в сеть с десктопа, на 4% сократилось [1]. Однако в связи с характерными особенностями смартфонов проектировщикам и дизайнерам часто приходится искать новые возможности повышения пользовательского опыта в приложениях. При этом оценка самого продукта также осложняется множеством факторов, начиная от недостатков подходов к тестированию и заканчивая трудной реализацией конкретных методов внутри них.

Мобильные устройства по своей природе обладают такими чертами, как персонализация, повсеместность, дифференцированность и районный характер. При проектировании приложений наиболее общими требованиями к ним являются мобильность, удобство использования (юзабилити), интероперабельность (от англ. interoperability), возможность подключения, безопасность и конфиденциальность. Однако внутри перечисленных критериев разные типы приложений (нативные, гибридные и веб) будут существенно отличаться, поэтому при разработке дизайна, выборе методов тестирования и их аспектов необходимо учитывать эти особенности.

На сегодняшний день существует четыре основных подхода к тестированию мобильных приложений: тестирование на базе эмуляции (Android SDK), тестирование на базе устройств, облачное тестирование (NTT DATA) и краудсорсинг (uTest). Первый из них основан на использовании эмулятора (симулятора) мобильного устройства, который

имитирует его поведение в виртуальной машине на персональном компьютере. Метод не требует вложений в создание лаборатории тестирования, но может применяться только для оценки функциональности системы, так как эмулятор не способен распознать полную палитру жестов (от англ. gestures) и создается лишь под конкретный смартфон или платформу. Тестирование на основе устройства хорошо справляется с проверкой качества обслуживания, однако плохо реагирует на быстрые изменения платформ и устройств и не подходит для масштабных исследований. В этом случае себя хорошо зарекомендовало облачное тестирование: сервисы предоставляют пользователю аренду необходимой тестовой среды и инструментов автоматизации для качественных крупных исследований, что не может гарантировать краудсорсинг [2].

Отдельно стоит остановиться на наиболее важном для UX аспекте тестирования – юзабилити. Оценка разрабатываемого продукта чаще всего производится по трем пунктам: системные характеристики (цель, юзабилити, функциональность, комплексность), внутреннее состояние пользователя (восприятие, настроение, желания, потребности) и контекст использования (организационный, культурный, социальный, временной, физический). Наиболее частыми проблемами при проведении подобных исследований для мобильных приложений являются большой разброс выделенных критериев в количественных методах, распространенных в сети, а также риск двусмысленной трактовки человеческих реакций, на которых базируются качественные методы.

Примером хорошего количественного метода может послужить опросник Т. Кескинена SUXES, в основе которого лежит маркетинговый подход по определению качества обслуживания и пирамида из шести элементов опыта и пяти уровней его глубины. Преимущество данного метода в его универсальности: он может применяться практически к любому продукту, а также допускает включение дополнительных необходимых критериев [3]. Самыми используемыми качественными методами остаются наблюдение и интервью.

Особенности подходов к повышению пользовательского опыта в приложениях напрямую связаны с небольшим размером экрана и разрешением мобильных устройств, а также их сенсорной природой. Так, главными принципами при разработке дизайна приложений становятся минималистичный UI, пользовательское поведение (реакция приложения на прикосновения пальцев), первичность контента; однако они дают слишком расплывчатое представление о том, каким должен быть продукт, чтобы соответствовать ожиданиям потребителей.

Чаще всего от неправильного проектирования взаимодействия страдают такие элементы интерфейсов приложений, как формы, навигация, потоки задач, списки, большие данные, обратная связь, аутентификация и т.п. Здесь имеет смысл руководствоваться не общими требованиями без четких формулировок, а рассматривать частные случаи, как это сделали П. Боровски и Т. Лауринавичюс в книге «Мобильный дизайн» [4]. Авторы предложили готовые решения для повышения пользовательского опыта с примерами их хорошего и плохого применения.

Другим возможным способом улучшения UX является использование новых технологий, как, например, вышеупомянутые жесты и FaceID, впервые примененная в iPhone X. Она предполагает идентификацию пользователя путем создания структурной карты лица с помощью системы камер TrueDepth, что обеспечивает при входе в систему нулевую стоимость взаимодействия, и пользователю не нужно запоминать рисунок или пароль. Но пока что нововведение уступает отпечатку пальца в скорости и необходимости использования только при определенном освещении и ракурсе. Жесты как основное средство «общения» с сенсорным экраном активно развиваются, и сейчас существуют как основные движения (tap, touch и т.д.), так и их интерпретации (touch and hold, pinch it open и т.д.). Жесты невидимы, что является их главным преимуществом и одновременно недостатком: с одной стороны, экономия ценного экранного места, а с другой – пользователям может быть трудно их запоминать в связи с многозначностью на различных устройствах и платформах

[5]. В связи с этим многие исследователи ссылаются на необходимость создания единого словаря жестов.

Таким образом, главными проблемами современных подходов к оценке мобильных продуктов остаются отсутствие конкуренции у облачных сервисов в инструментарии и масштабности, из-за чего тестирование возможностей приложений (например, совместимости, кроссплатформенности) становится труднодоступным, а также отсутствие единых критериев в тестированиях юзабилити и недостаточная точность качественных исследований. Для улучшения пользовательского опыта в приложениях также необходимо сформулировать более конкретные требования к проектированию интерфейсов.

Литература

1. Количество пользователей интернета в России [Электронный ресурс]. – Режим доступа: http://www.bizhit.ru/index/users_count/0-151, своб.
2. Gao J., Bai X., Tsai W.-T., Uehara T. Mobile Application Testing: A Tutorial [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/260603816_Mobile_Application_Testing_A_Tutorial, своб.
3. Keskinen T. Evaluating the User Experience of Interactive Systems in Challenging Circumstances [Электронный ресурс]. – Режим доступа: <http://tampub.uta.fi/bitstream/handle/10024/98132/978-951-44-9972-2.pdf;sequence=1>, своб.
4. Borowska P., Laurinavicius T. Mobile Design Book [Электронный ресурс]. – Режим доступа: <http://mobiledesignbook.com/>, орг.
5. Budiu R. The State of Mobile User Experience [Электронный ресурс]. – Режим доступа: <https://www.nngroup.com/articles/state-mobile-ux/>, своб.



Полякова Дарья Александровна

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: pooda6a@gmail.com



Сокуренок Юрий Андреевич

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», к.т.н., доцент

e-mail: kpd@limtu.ru

УДК 004.356.2

АНАЛИЗ ЗАРУБЕЖНЫХ И ОТЕЧЕСТВЕННЫХ ИССЛЕДОВАНИЙ ПРОБЛЕМЫ АДАПТАЦИИ 3D-МОДЕЛЕЙ ДЛЯ 3D-ПЕЧАТИ

Полякова Д.А.

Научный руководитель – к.т.н., доцент Сокуренок Ю.А.

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

Подведен итог работы по анализу зарубежных и отечественных исследований методов адаптации 3D-моделей для 3D-печати и классификацией дефектов 3D-печати, связанных с некорректной подготовкой 3D-моделей.

Ключевые слова: адаптация 3D-моделей, дефекты 3D-печати, подготовка 3D-моделей к печати, FDM-технология, 3D-принтер.

Несмотря на широкие возможности 3D-печати и стремительный рост доступности 3D-принтеров, далеко не каждую 3D-модель, полученную из сети Интернет легко адаптировать для последующей печати [1]. Даже самый простой геометрический примитив требует тщательного рассмотрения на допустимость печатных размеров, соразмерность габаритам печатной камеры, возможность применения печатной технологии, а также соответствие материалов. С другой стороны, процесс моделирования объекта в программной среде «с нуля» не решает вышеперечисленные проблемы, а лишь усугубляет сложившуюся ситуацию. В иностранных источниках рассмотрена проблема кроссплатформенности слайсеров, авторами подчеркивается проблема игнорирования пользователями правил использования программного обеспечения. Использование программного обеспечения, предназначенного для 3D-скульптинга, нецелесообразно для моделирования печатного объекта. Программы такого спектра создают особые оболочки, увеличивают размер файла и количество полигонов, что затрудняет работу программ-слайсеров.

Для решения проблем адаптации 3D-моделей специалистами рекомендуется использование программ отслеживания ошибок, таких как Autodesk Netfabb и Autodesk Meshmixer. Однако данные программы не могут решить проблемы искажения поверхности или несоответствия форме первоначального объекта, полученного, например, методом 3D-сканирования или фотограмметрии. Такие 3D-модели не могут быть обработаны только в

автоматическом режиме и требуют ручного вмешательства и доработки. Неадаптированные 3D-модели крайне негативно влияют на эргономику и производительность производства. Невостребованный испорченный пластик, который не будет в дальнейшем переработан и энергия, затраченная на работу принтера впустую, заставляют задуматься о влиянии печатного производства на окружающую среду. На сегодняшний день отсутствует четкая методология подготовки 3D-моделей для 3D-печати. Возникает противоречие между растущей популярностью 3D-принтеров, а также разнообразием печатных технологий с одной стороны, с трудностями технической и программной реализации 3D-моделей для 3D-печати с другой. Возможные дефекты 3D-печати представлены на рисунке.

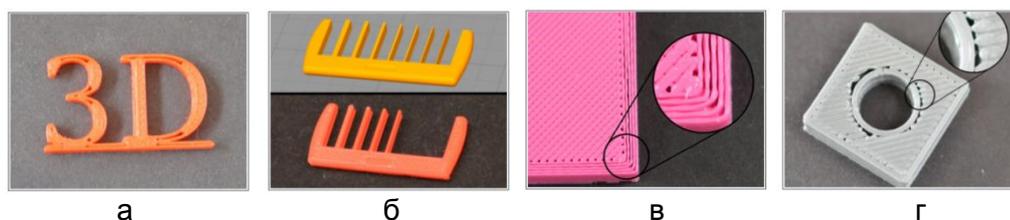


Рисунок. Примеры дефектов 3D-печати, возникших по причине неправильной подготовки 3D-моделей: разрывы (а); отсутствие элементов (б); закругленные углы (в); щели (г)

Перед отправкой 3D-модели на печать ее необходимо адаптировать по ряду конкретных критериев в первую очередь зависящих от выводного устройства. Согласно исследованиям авторов [2] можно выделить следующие критерии сравнения 3D-устройств для печати:

1. технология печати;
2. рабочий материал (филамент);
3. размеры рабочего пространства;
4. точность позиционирования, А;
5. толщина слоя, Т.

При моделировании мелких деталей необходимо обратить внимание на допустимую детализацию. Для корректной печати необходимо соблюсти минимальные значения основных параметров, которые зависят от используемого материала. В таблице приведены сравнительные характеристики материалов для 3D-печати.

Таблица. Минимальные значения параметров геометрии для материалов

Материал	Значение, мм					
	Толщина стенки	Толщина перемычки	Высота рельефа	Величина зазора	Толщина полки	Диаметр отверстия
Фотополимер	0,8	0,8	0,5	0,5	0,8	10
Полиамид	0,8	0,8	0,5	0,5	1	2
Латунь	0,6	0,8	0,3	0,3	0,6	4
Серебро						
Бронза						
Золото						
Титан	1	1	1	3	1	2
Сталь						
Гипс	2	2	0,8	1	2	15
Воск	0,5	0,5	0,3	0,5	0,5	1

На втором этапе подготовки моделей, особое внимание стоит уделить исправлению следующих элементов [3]:

1. провести проверку сетки объекта на отсутствие «открытых» ребер и пересекающихся треугольников. Полигональная сетка должна быть «герметичной», в противном случае 3D-принтер не будет воспринимать модель как твердотельный объект;
2. сложные или крупные модели необходимо разделить на составные элементы. Во-первых, печать сложной цельной детали повлечет за собой создание поддержек и всех сопутствующих недостатков, которые можно избежать при разбиении детали на части. Во-вторых, сборные элементы обладают большей прочностью и устойчивостью;
3. задать положение модели на рабочем пространстве, руководствуясь правилом «совокупная скорость печати в высоту всегда ниже скорости печати самих слоев» [3]. Оптимизация положения модели уменьшит количество поддерживающих структур;
4. провести проверку толщины стенок. Слишком тонкая стенка будет проигнорирована принтером, а слишком толстая – создаст слишком много внутреннего напряжения и может привести к трещине или даже сломать предмет.

Процесс моделирования и трехмерной печати во многом зависит от класса создаваемой модели. В исследованиях отечественных (Е.К. Карпов, И.Е. Карпова, В.В. Иванов и др.) ученых проблема адаптации трехмерных моделей для 3D-печати рассматривается в контексте разработки алгоритмов 3D-моделирования сложных классов: резьбовых и зубчатых соединений, шестерен, эвольвент [4]. При моделировании данных изделий особое внимание уделяется повышению износостойкости и прочности детали за счет ручного уплотнения полигональной сетки в местах наибольшего трения.

Прежде чем печатать модель, необходимо удостовериться, что имеется достаточный зазор между передачами, зубцами и ссылками в цепочках. Иначе прототип окажется сплошным, статичным объектом. Для добавления зазоров или отверстий возможно использование программы AutoCAD. При создании отверстий указывается тип места размещения, размеры отверстия, параметры сверления и варианты типа резьбы.

Сообщество разработчиков 3D-моделей в статье «Designing for 3D printing» выделяют несколько «узких мест» 3D-печати, на которые следует обратить внимание при проектировании 3D-моделей [5].

1. Закругленные углы. Поскольку сопло принтера круглое, невозможно создать совершенно резкий внешний угол на объекте. Даже если угол создан абсолютно прямым в CAD, 3D-принтер не способен создать такой край, при этом присутствует материальное наращивание. Если конфигурация модели позволяет скруглить углы на этапе проектирования, то результат печати будет более предсказуемым.
2. Фаски. Самый простой способ соединения частей – склейка. Однако возможности 3D-принтера позволяют оптимизировать данный процесс. Авторы рекомендуют использовать всевозможные штифты, отверстия. Круглые отверстия и штыри помогут добиться структурной целостности элементов. Для облегчения стыковки элементов на поверхность добавляют фаски. Также использование фасок обеспечивает уменьшение «точек стресса». Резкие переходы между прямыми формами и телами вращения рекомендуется компенсировать за счет построения фасок.
3. Правило 45°. При разработке модели нависающие углы должны быть расположены максимум до 45°. Принтер способен печатать углы круче, однако качество поверхности от этого становится хуже.
4. Скос. При создании «мостиков» исследователи не рекомендуют использовать скос. Медленный наклонный скос, приведет к довольно уродливому качеству поверхности в его нижней части. Используя прямой мост, значительно можно улучшить качество печати. Если требуется наклонный скос, необходимо использовать поддержку для печати, однако это приведет к необходимости грубой обработки поверхности.

5. Отверстия. Горизонтальные отверстия в модели вызывают сильный выступ на вершине арки. Это означает, что, как правило, горизонтальное отверстие будет немного неровным сверху и, следовательно, меньшим размером, чем предполагалось. Предлагаемые методы борьбы: изменить форму отверстия, сделав отверстие более похожим на каплю воды. Это облегчит работу принтера, создается допуск погрешности. Второй подход: создать тонкую опорную мембрану, чтобы удерживать верхнюю часть отверстия. Вместо того, чтобы покрывать все отверстие сеткой, необходимо смоделировать узкую опору в нижней части и наращивать слои к вершине арки.

Уменьшение диаметра центрального отверстия целесообразно избежать благодаря моделированию отверстий диаметром примерно на 0,005 см больше требуемого.

Таким образом, 3D-модель должна быть подготовлена с учетом структурных особенностей геометрии, специфики технологии 3D-принтера и выбранного материала. Дальнейшее направление работы включает в себя создание четкой классификации среди методов и средств подготовки 3D-моделей, а также разработку алгоритмов трехмерного моделирования, ориентированного на трехмерную FDM-печать.

Литература

1. Шалобаев Е.В., Перепелица Ф.А., Красноруцкая Н.С. Аддитивные технологии в машиностроении // В сб.: Приборостроение в XXI веке – 2016. Интеграция науки, образования и производства. Материалы XII Международной научно-технической конференции. – 2017. – С. 319–323.
2. Гуцин И.А., Авдеев А.Р., Швец А.А., Дроботов А.В. Принципы создания программы работы устройства объемной печати по трехмерной модели // Изв. ВолгГТУ. – 2015. – № 11(173). – С. 50–53.
3. 10 правил подготовки модели к 3D-печати [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/196182/>, своб.
4. Карпов Е.К., Карпова И.Е., Иванов В.В. Особенности моделирования резьбовых соединений в САД-системах для последующей 3D-печати, на примере Компас-3D // Зауральский научный вестник. – 2014. – № 1(5). – С. 25–27.
5. Designing for 3D printing [Электронный ресурс]. – Режим доступа: <http://support.3dverkstan.se/article/38-designing-for-3d-printing/>, своб.



Романова Асель

Год рождения: 1993

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, аспирант

Направление подготовки: 09.06.01 – Информатика
и вычислительная техника

e-mail: asel-romanova@mail.ru

УДК 004.946

ПЕРСПЕКТИВЫ РАЗВИТИЯ УСТРОЙСТВ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

Романова А.

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

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

В работе рассмотрены перспективы развития физических устройств, поддерживающие работу в виртуальной среде. Произведен анализ существующих проблем, препятствующих динамичному развитию технологии.

Ключевые слова: виртуальная реальность, устройства виртуальной реальности, погружение, кибернетика.

Развитие информационно-коммуникационных технологий (ИКТ) в настоящее время обусловлено высоким спросом на рынке, виртуальная и дополненная реальность относится к одной из наиболее динамично развивающейся отрасли во всем мире. Зарождение виртуальных технологий образовалось в 1960 году благодаря Мортону Хейлингу [1], представивший игровую приставку «Sensorama», конструкция представляла собой движущее кресло, колонки для воспроизведения звука и экран, показывающий трехмерное изображение. Однако для того времени данное устройство не пользовалось большим спросом, настоящий скачок развития в области виртуальной реальности (VR) произошел в конце XX века. Основной проблемой тормозящей развитие на протяжении всего периода существования виртуальных технологий являлась недостаточная мощность вычислительных машин [2], недостаточное финансирование и непонимание технологии целевой аудиторией.

Следует отметить ряд проблем, существующих на данный момент: высокая стоимость шлемов и очков виртуальной реальности, для качественного погружения в среду VR необходимо обладать достаточно мощным по производительности компьютером и мобильным телефоном, а также не забывать о важности передачи цвета. Существует мнение, что дороговизна устройств в ближайшие время будет существенно снижаться за счет конкуренции и разработки достаточно производительной модели для широкого круга пользователей по доступной цене.

Следующей проблемой развития виртуальных технологий отмечается наличие проводов, мешающих комфортному времяпровождению в игровой среде, где требуется постоянное движение и работа с контроллерами. Решение данной проблемы представила только компания HTC Vive [3], спроектировав в этом году модуль беспроводного подключения к источнику сигнала под названием TPCAST Wireless Adaptor for Vive. Обновленный аксессуар на рисунке использует TX Модуль (отправляет сигналы и получает обратную связь), RX Модуль (получает сигнал от TX модуля и транслирует эту картинку на экран шлема) и батарею (с безопасными и эффективными аккумуляторами, адаптер сможет работать от 2 до 5 ч). Таким образом, компании подбираются к тому самому устройству,

которое в итоге сможет объединить все наработки, к которым сообщество шло годами и в скором будущем представить шлем, избавленный от большинства недостатков и проблем современных моделей.



Рисунок. Шлем TPCAST Wireless Adapter for VIVE

Невысокая глубина погружения и короткая длина игровой сессии являются еще одним недостатком физических устройств. Решение данной проблемы будет решаться годами, если не десятилетиями в процессе эволюции оборудования виртуальной реальности. Существует еще ряд проблем, препятствующий быстрому развитию VR: разрешение экрана, плотность пикселей, вес оборудования, поддержка на всех платформах, встраивание рекламы и монетизация. В связи с этим перспектива развития устройств виртуальных технологий медленно, но движется вперед. Со времен запуска технологии, пик завышенных ожиданий был несколько велик по сравнению с возможностями самого оборудования, поэтому это привело к спаду интереса среди пользователей, однако существующие решения вновь возвращают заинтересованность, в конечном счете, это приведет к постоянному плато производительности.

Литература

1. Сенсорам Мортон Хейлинга [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Сенсорам>, своб.
2. Романова А., Шуклин Д.А. Проблемы и перспективы применения виртуальной реальности в высшем образовании // Современное образование: традиции и инновации. – 2017. – № 1. – С. 51–54.
3. Рассуждения на тему виртуальной реальности [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/349470/>, своб.



Ростов Данил Сергеевич

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: daniil.rossy@gmail.com



Сокуренок Юрий Андреевич

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», к.т.н., доцент

e-mail: kpd@limtu.ru



Государев Илья Борисович

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: goss@corp.ifmo.ru

УДК 004.04

**МЕТОДЫ ВЫБОРА ПРОГРАММНЫХ СРЕДСТВ РАЗРАБОТКИ В ЗАВИСИМОСТИ
ОТ ВЕБ-ИНТЕРФЕЙСА**

Ростов Д.С., Сокуренок Ю.А., Государев И.Б.

Научный руководитель – к.т.н., доцент Сокуренок Ю.А.

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

В работе проведен анализ взаимосвязи между методами выбора программных средств разработки и веб-интерфейсами в проекте. Рассмотрены исследования по изучению фундаментальных принципов выбора методологий разработке IT-продуктов, методов проектирования веб-интерфейсов и подходов к выбору технологий для разработки.

Ключевые слова: разработка веб-сайтов, инструменты разработки сайтов, веб-интерфейс.

На сегодняшний день веб-проекты могут существенно отличаться друг от друга. Это может быть как одностраничный сайт для продажи конкретного товара, так и многофункциональная система взаимодействия большого количества людей друг с другом, имеющая крупную базу данных и сложную логику обработки поступающей от пользователей информации.

Актуальность темы обосновывается тем, что по мере развития и внедрения новых технологий, переосмысливаются и подходы к веб-разработке. Одновременно происходят существенные изменения в использовании веб-сайтов. Изменения технологий инициируют изменения в поведении пользователей на веб-страницах. Новые технологии позволяют информационным системам генерировать более эффективные сценарии взаимодействия [1].

С одной стороны, технологии открывают новые возможности для проектирования веб-интерфейсов. С другой стороны, требования к современным веб-интерфейсам постоянно растут и тем самым заставляют разработчиков совершенствовать методы разработки и подстраиваться под стоящие перед ними задачи.

Для того чтобы понять, какая существует взаимосвязь между методами выбора программных средств разработки и веб-интерфейсами, следует рассмотреть основные принципы разработки программного продукта [2].

При разработке программного продукта можно руководствоваться различными методологиями. Выбор методологии может зависеть от многих факторов: специфика проекта, бюджет, поставленные задачи, субъективные предпочтения и еще многие детали.

В данной работе рассмотрены 7 методологий: «Водопад», «Model-V», «Инкрементная модель», «RAD»-модель», «Agile-модель», «Итерационная модель», «Спиральная модель» [3]. Каждая из методологий предполагает разные взаимоотношения внутри команды в проекте. К примеру, при разработке простых сайтов чаще используют «Водопад», так как эта методология проста во внедрении. Ее суть заключается в том, что каждый этап разработки начинается после того, как завершен предыдущий, т.е. приступить к программной реализации возможно только после окончания проектирования веб-интерфейса. Однако при использовании гибких методологий появляется возможность вносить корректировки в любую часть проекта и на любой стадии. Используя «Водопад», разработчики будут уверены, что от них не потребуются реализации с существенными отличиями от первоначального варианта, что это будет учтено при выборе технологий. Если же предполагаются изменения, то необходимо будет использовать те технологии, которые обеспечат хорошую масштабируемость и гибкость проекта. Это значит, что в зависимости от выбранной методологии, будет существовать различная взаимосвязь между используемыми в разработке технологиями и веб-интерфейсом. Это зависит от того, на каких стадиях разработки будет происходить проектирование веб-интерфейса.

По мере развития веб-технологий меняются и подходы к проектированию веб-интерфейсов. Основной задачей проектирования веб-интерфейсов является генерация целевых действий пользователя. Для того чтобы взаимодействие пользователя с информационной системой было максимально продуктивным, нужно учитывать множество факторов (цель его прихода на определенную веб-страницу, устройство, с помощью которого пользователь загружает веб-страницы, производительность его устройства, скорость интернет-соединения и много другое) [4].

В последнее время веб-технологии быстро развиваются. Это тесно связано с частым появлением новых версий браузеров и поддержкой ими новых технологий. Из этого следует, что направления веб-разработки постоянно изменяются. Часто случается так, что появление новой технологии порождает новые методы разработки, новые методы выбора инструментов разработки и переосмысления всего процесса разработки.

Выбор технологий для веб-проекта на сегодняшний момент является одним из самых значимых этапов разработки. От правильности выбора программных решений зависит множество показателей: сроки выполнения, трудоемкость, бюджет, успешность реализации, качество продукта.

Проблема заключается в субъективности выбора. Для оптимального выбора технологий человек или команда должны понимать специфику проекта, иметь многолетний опыт разработки, владеть различными языками разработки, быть знакомыми с построением аналогичных проектов [4].

При разработке простых проектов часто встает вопрос выбора CMS (Content Management System). Выбор CMS зависит от поставленной задачи, которую будет решать эта система: специфика сайта, уровень квалификации разработчиков, уровень квалификации бедующих модераторов проекта, формат поддержка и масштабирования, необходимый уровень производительности [5].

При разработке средних проектов (крупные интернет-магазины, порталы, сервисы и сложные приложения и т.д.) рационально использовать веб-фреймворки. В таких проектах чаще всего требуется компонентный подход, реактивность данных и сложный, интерактивный интерфейс. Фреймворки предоставляют для этих задач готовые структуры кода в упрощенной форме, позволяя не писать всю низкоуровневую структуру вручную.

Для разработки очень сложных проектов с требованиями к повышенной гибкости, нагрузке рационально использовать чистые языки программирования. Чем больше проект, тем больше требований. Фреймворки же хоть и позволяют разрабатывать сложные интерфейсы и реализовывать крупную логику, имеют определенные ограничения. Каждый язык программирования подходит для решений определенной задачи. Более того, разные части проекта могут работать на разных технологиях и языках [2].

В ходе исследования были рассмотрены работы по изучению фундаментальных принципов выбора методологий при разработке IT-продуктов, методов проектирования веб-интерфейсов и подходов к выбору технологий для разработки.

Литература

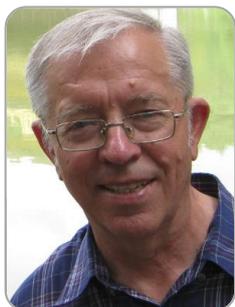
1. Азимова Д.Ю., Турсунова Д.Г. Современный подход к разработке веб-ресурсов // Молодой ученый. – 2016. – № 21. – С. 745–747.
2. Вигерс К., Битти Д. Разработка требований к программному обеспечению. – 3-е изд., доп. / Пер. с англ. – М.: Русская редакция; СПб.: БХВ-Петербург, 2014. – 736 с.
3. Еще раз про 7 основных методов разработки [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/company/edison/blog/269789> (дата обращения: 10.12.2017).
4. Henderson С. Building scalable web sites. – O'Reilly, 2006. – 352 p.
5. How to choose right CMS [Электронный ресурс]. – Режим доступа: <https://www.templatemonster.com/blog/how-to-choose-the-right-cms-for-my-website> (дата обращения: 11.12.2017).

**Савенкова Ирина Алексеевна**

Год рождения: 1991

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207сНаправление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: irina.savenkova@hotmail.com

**Погорелов Виктор Иванович**

Год рождения: 1940

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, д.т.н., профессор

e-mail: kpd@limtu.ru

УДК 004.558

**ОСОБЕННОСТИ РАЗРАБОТКИ ГЕЙМИФИЦИРОВАННОГО ПРИЛОЖЕНИЯ
ДЛЯ ОПЕРАЦИОННОЙ СИСТЕМЫ ANDROID****Савенкова И.А.****Научный руководитель – д.т.н., профессор Погорелов В.И.**

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

В работе рассмотрено сочетание игровой и полезной компонент при создании мобильного геймифицированного приложения. Выявлены достоинства и недостатки операционной системы Android, кратко рассмотрены основные компоненты Android и последовательность разработки приложения для данной платформы. Предложено эффективное сочетание игровых элементов при создании обучающего мобильного геймифицированного приложения.

Ключевые слова: игрофикация, геймификация, Android, игровые технологии, образовательные игры.

За последние годы мобильные телефоны из средств связи трансформировались в многофункциональные устройства. Они совмещают в себе функции плеера, видеопроигрывателя, игровой консоли, устройства для выхода в Интернет, камеры, электронной книги и т.д. Мобильные приложения предоставляют пользователям значительные возможности персонализации функциональности смартфона. Также можно констатировать, что мобильные геймифицированные приложения и образовательные игры пользуются популярностью в магазинах приложений.

В основе геймифицированного приложения, равно как и образовательной игры, лежит сочетание игровой и полезной компонент. Далее данные компоненты рассмотрены применительно к средствам мобильной связи. Первая игра для мобильных телефонов – «Змейка» – появилась в 1997 году на телефонах компании Nokia. Игра быстро завоевала внимание клиентов компании, что побудило конкурентов Nokia устанавливать игры на своих моделях. В настоящее время игры сильно эволюционировали, неизменным остается только интерес к ним пользователей всех возрастов.

Полезной компонентой является помощь в соблюдении диеты, отказе от курения, соблюдении стандартов обслуживания и т.д. Применительно к обучению полезной

компонентой является образовательный контент, который позволяет развить компетенции учащегося. Стоит отметить, что образовательные приложения появились сравнительно недавно, их популяризации способствовала глобализация, а также ускорение технического прогресса, в результате чего в обществе стала доминировать идея непрерывного обучения, т.е. обучения на протяжении всей жизни, в отличие от прежнего подхода, при котором человек, пройдя обучение в молодом возрасте, не уделял значительного внимания дальнейшему углублению знаний.

Для обеспечения одновременного достижения игровой цели и цели обучения необходимо, чтобы игровые и обучающие задания выполнялись одновременно, т.е. были объединены в одно действие. Это позволяет обеспечивать баланс игровой и обучающей компоненты и сохранять таким образом целостность восприятия игры и достигать цели обучения. Широкое распространение получили приложения для изучения иностранного языка, программирования, улучшения когнитивных функций, развития творческих навыков.

В настоящее время мобильные приложения создаются преимущественно для двух платформ: iOS и Android. В связи с тем, что 80% устройств функционируют на базе Android, в работе сделан акцент на данной операционной системе (ОС). Е.В. Куприй [1] указывает, что очевидными достоинствами Android являются гибкость, открытые исходные коды, множество программ, высокое быстродействие, удобное взаимодействие с сервисами Google, многозадачность. Можно добавить, что при разработке приложений основным преимуществом Android по сравнению с iOS также являются менее строгие требования к публикации приложения. А основным недостатком – необходимость поддержки большого числа устройств, разрешений экранов, версий ОС.

Android – операционная система, которая была создана для работы на смартфонах, планшетных компьютерах, электронных книгах, цифровых проигрывателях, «умных» наручных часах, игровых приставках, нетбуках, смартбуках, очках Google, телевизорах, системах автоматического управления автомобилем и других устройств. ОС создана на основе ядра Linux и виртуальной машины Java. Android SDK включает в себя разнообразные библиотеки, документацию и инструменты, которые помогают разрабатывать мобильные приложения для платформы Android.

API Android SDK – API-библиотеки Android, предоставляемые для разработки приложений. Документация SDK включает обширную справочную информацию, детализирующую, что включено в каждый пакет и класс, и как это использовать при разработке приложений.

Полезным средством при разработке приложений под Android является AVD (Android Virtual Device) – интерактивный эмулятор мобильного устройства Android. Используя эмулятор, можно запускать и тестировать приложения без использования реального Android-устройства. Как отмечено ранее, недостатком Android является необходимость поддержки большого числа устройств, AVD частично решает данную проблему, позволяя протестировать, будет ли приложение правильно работать и отображать информацию на различных устройствах. Тем не менее, первоначальную разработку удобнее осуществлять с использованием реального устройства.

Далее перечислены основные компоненты Android.

Activity (активность, деятельность) – представляет собой схему представления Android-приложений. Каждый экран пользовательского интерфейса представлен классом Activity. Android-приложение способно состоять из нескольких форм (Activity) и может переключаться между ними во время выполнения приложения.

Intents (намерения) – асинхронные сообщения, которые позволяют приложению запросить функции из других служб или действий. Приложение может делать прямые запросы службе или действию (явное намерение) или запросить у Android зарегистрированные службы и приложения (неявное намерение).

Views (представление, вид, элемент управления) – пользовательский интерфейс, создаваемый виджетами классов. Сюда относятся кнопки, текстовые поля, флажки, переключатели и т.п.

Services (службы) выполняет фоновые задачи без предоставления пользовательского интерфейса (например, проигрывание музыки), может уведомлять пользователя через систему уведомлений Android.

Content Provider (контент-провайдер) предоставляет данные приложениям. С помощью контент-провайдера приложение может обмениваться данными с другими приложениями. Android содержит базу данных SQLite, которая может выполнять роль контент-провайдера.

Broadcast Receiver (приемник широковещательных сообщений/запросов) принимает системные сообщения и неявные намерения, может использоваться для реагирования на изменение состояния системы. Приложение может регистрироваться как приемник определенных событий и может быть запущено, если такое событие произойдет.

Другими компонентами Android являются виджеты, или живые папки (Live Folders), или живые обои (Live Wallpapers). Живые папки отображают источник любых данных на «рабочем столе» без запуска соответствующих приложений.

Последовательность разработки приложения включает в себя знакомство с языком программирования Java (в последнее время популярность приобретает язык Kotlin) и платформой, создание APK-файла для распространения, публикацию приложения, его продвижение и монетизацию.

Образовательное приложение может быть выполнено как классическая геймификация, т.е. приложение, состоящее из теоретической части, за освоение которой присваиваются очки и бейджи, и тестовой части, в которой игроку предлагается проверить полученные знания. С другой стороны, весьма полезным может оказаться большой уклон в образовательные игры, при которых игрок погружается в процесс, а акцент сделан на практике.

Независимо от того, как будет выглядеть итоговое мобильное обучающее приложение, автор полагает, что повысить его эффективность способны следующие элементы: конкуренция, личный прогресс, выбор траектории обучения, быстрая обратная связь, возможность получения помощи, отсутствие наказания за ошибку.

Конкуренция является одним из основных элементов, позволяющих улучшить результаты. Среди пользователей можно проводить как регулярные (например, ежедневные) соревнования, так и разовые марафоны. Призы могут быть виртуальными (бейджи, бесплатное продление подписки и т.д.) и реальными. Для определения победителя прекрасно подходит такой элемент геймификации как очки. Как отмечают исследователи В.И. Погорелов, Д.А. Шуклин, Д.В. Зимина, О.О. Козак [2], главная цель очков – поощрение движения пользователя по «правильному» пути, но существует и дополнительное назначение – использование как ресурса, который обменивается на другие ресурсы или ценные призы, показателя статуса, отношения к игроку со стороны его товарищей и, в определенном количестве, – состояния достижения победы.

Наряду с общим зачетом можно выделять соревновательные группы, состоящие из двух и более человек. Группы могут как состоять из случайных игроков, так и подбираться в зависимости от уровня или других параметров. Интересным вариантом конкуренции представляется возможность авторизации через социальные сети таким образом, чтобы игроки всегда видели результаты, достигнутые пользователями из их списка контактов.

Личный прогресс также играет важную стимулирующую роль, особенно в том случае, если игрок на текущий момент не достаточно успешен в конкуренции. Таким образом, даже не входя в топ игроков, пользователь остается замотивированным, наблюдая личный прогресс, стремясь улучшить индивидуальные результаты. Предпочтительно разработать систему подробной статистики для различных временных промежутков, а также учитывающую различные метрики.

Выбор траектории обучения возможен не всегда, однако является достаточно сильным мотивирующим фактором. Например, режимы игры могут значительно разнообразить игровой опыт.

Быстрая обратная связь чрезвычайно важна. Быструю обратную связь можно обеспечить посредством автоматизированных инструментов, которые могут оценить правильность решения. Сейчас такие инструменты применяются в виде тестов с вариантами ответов, а также широко используются при обучении программированию, когда тестируется правильность выходных данных при выполнении программы путем подачи на вход различных входных данных (как правило, неизвестных пользователю). Кроме того, можно адаптировать для мобильных приложений систему peer review, часто используемую в МООС (Massive Open Online Course).

Возможность получения помощи очень важна. Причем возможность получения помощи может быть реализована различными способами: автоматическая подсказка от программы, помощь ментора, специальный раздел на сайте, образовательное сообщество.

Отсутствие наказания за ошибку – пользователь не должен бояться совершить ошибку (если только он сознательно не выбрал опцию безошибочности в качестве режима игры в вышеупомянутой траектории обучения). На взгляд автора, пользователь должен иметь возможность ошибаться без потери баллов (например, при решении задачи, если цель уровня – решение задачи или прохождение теста) либо с потерей баллов, равноценных полученным за правильный ответ (в случае, если вариант без потери баллов неприменим в связи с тем, что случайный выбор вариантов позволит набирать больше баллов, чем осознанный выбор).

Технические возможности современных мобильных телефонов на базе Android позволяют создавать приложения, обеспечивающие поддержку всех вышеперечисленных элементов. В то же время нужно помнить не только о внутреннем содержании, но и о представлении. Как справедливо замечено в работе [3], визуальный дизайн играет важную роль. Причем визуальный дизайн должен быть как эстетически привлекательным (UI), так и функциональным (UX).

Таким образом, можно сделать вывод, что мобильные приложения активно используются в современном мире, а платформа Android является лидирующей на рынке. В то же время пользователи заинтересованы в игровом обучении. В связи с этим предложено эффективное, на взгляд автора, сочетание игровых элементов, включающих в себя конкуренцию, личный прогресс, выбор траектории обучения, быструю обратную связь, возможность получения помощи, отсутствие наказания за ошибку.

Литература

1. Куприй Е.В. Особенности операционной системы Android // Модели, системы, сети в экономике, технике, природе и обществе. – 2012. – № 3. – С. 139–143.
2. Шуклин Д.А., Погорелов В.И., Зимина Д.В., Козак О.О. Особенности использования элементов геймификации в системах дистанционного обучения // Технические науки. Теория и практика: материалы II международной научно-практической конференции. – 2017. – С. 68–72.
3. Колобаева А.А., Погорелов В.И., Шуклин Д.А. Внутренняя геймификация как способ повышения эффективности сотрудника // Успехи современной науки и образования. – 2017. – Т. 4. – № 3. – С. 174–177.

**Севостьянова Юлия Сергеевна**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4107Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: juliashka3010@mail.ru

**Шуклин Дмитрий Анатольевич**

Год рождения: 1979

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: do@limtu.ru

УДК 004**АНАЛИЗ НАУЧНЫХ ИССЛЕДОВАНИЙ ПРОБЛЕМЫ ПРОЕКТИРОВАНИЯ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ВЗАИМОДЕЙСТВИЯ ПОСТАВЩИКОВ И ПОТРЕБИТЕЛЕЙ****Севостьянова Ю.С., Шуклин Д.А.****Научный руководитель – к.педагог.н., доцент Шуклин Д.А.**

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

В работе рассмотрены основные проблемы, с которыми сталкиваются веб-разработчики при проектировании веб-приложений, приведены требования, предъявляемые к веб-приложениям уровня предприятия. В заключении обсуждены способы устранения данных проблем как с точки зрения теории, так и с точки зрения практической реализации.

Ключевые слова: веб-приложение, этапы разработки, Rich Internet application, AJAX, производительность.

В настоящее время веб-технологии могут использоваться в различных сферах жизнедеятельности человека, например, для организации взаимодействия пользователей, продажи товаров и услуг, организации взаимодействия с клиентами. Разработка веб-приложений является достаточно трудоемким процессом, который включает анализ предметной области, изучений предпочтений заказчика, определение требований к программному продукту, проектирование интерфейса и выбор среды разработки, внедрение и сопровождение продукта. Каждый разработчик должен уметь проанализировать возможности сред и языков программирования согласно выделенным критериям, поскольку от этого зависит продолжительность и результативность работы приложения.

Неопровержимые данные и опыт показывают, что разработчики, пользователи и другие заинтересованные стороны по-прежнему сталкиваются с проблемами развития ad hoc. В результате этого возрастает озабоченность по поводу того, как создаются сложные веб-приложения, а также уровень производительности, качества и целостности этих приложений. Основными причинами этих неудач являются не технологии, а недальновидные цели, ошибочный дизайн и процесс развития [1].

При этом, также согласно исследованиям, часто встречаются такие проблемы, как устаревшая или нерелевантная информация, трудности с использованием веб-сайта и поиск

соответствующей интересующей информации, обслуживание, медленный отклик, масштабируемость, сбои веб-сайтов и нарушения безопасности. Веб-разработчики часто упускают из виду важные нетехнические соображения, такие как авторское право и конфиденциальность. Большинство веб-приложений также не имеют надлежащего тестирования, оценки и документации.

Современные многофункциональные веб-приложения должны отвечать требованиям, предъявляемым к веб-приложениям уровня предприятия [2]:

- наличие внешних (пользовательских) и внутренних (корпоративных) разделов сайта;
- единообразие внешнего вида всех генерируемых веб-страниц;
- высокая производительность обслуживания пользователей.

Чтобы успешно создавать сложные веб-приложения для взаимодействия поставщиков и потребителей, веб-разработчикам необходимо принять дисциплинированный процесс разработки и надежную методологию, использовать лучшие инструменты для разработки и следовать ряду хороших рекомендаций.

В настоящий момент существует множество технологий, реализующих логику веб-приложений на стороне сервера. Современные веб-приложения работают на распределенных аппаратных и гетерогенных компьютерных системах. Кроме того, благодаря новым возможностям беспроводных технологий и переносных вычислительных и коммуникационных устройств быстро развивается новая волна мобильных веб-приложений.

Появление Rich Internet application (RIA, «богатое Интернет-приложение») позволило расширить круг решаемых задач и расширить сферу применения веб-приложений. До появления технологии AJAX веб-приложения выполнялись преимущественно на стороне веб-сервера. Веб-браузер в таком случае играл роль пассивного монитора, который отображал полученный с веб-сервера HTML-документ. Управление поведением элементов пользовательского интерфейса осуществлялось путем полной перерисовки HTML-документа, полученного новым запросом с веб-сервера, а это слишком тяжеловесное решение [3].

Большинство веб-разработчиков, считает, что разработка веб-приложений – это создание веб-страниц с использованием программного обеспечения для веб-разработки HTML или Web, например, Front Page или Dreamweaver, и воплощение нескольких изображений и гиперссылок документов и веб-страниц. Хотя к этой категории относятся некоторые простые приложения, такие как персональные веб-страницы, анонсы семинаров и простые онлайн-брошюры компаний, которые требуют простой презентации и навигации контента, многие веб-приложения сложны и должны отвечать множеству сложных требований, которые меняются и развиваются [3].

Е.В. Золотарев поясняет, что работа над приложением включает несколько этапов, соответствующих следующим этапам жизненного цикла [4]:

- определение требований к приложению;
- проектирование приложения;
- разработка;
- тестирование проекта;
- внедрение проекта;
- сопровождение проекта.

За последние несколько лет появилось множество веб-технологий и стандартов для лучшей поддержки новых веб-приложений: XML, веб-сервисов, Semantic Web, веб-технологий персонализации, веб-разработки, веб-аналитики и мобильных и контекстно-зависимых сервисов. Достижения в веб-технологиях и преимущества, которые они предлагают, привели к разнообразному спектру приложений.

Анализ исследования [3] показал, что для достижения максимальной производительности веб-приложения необходимо минимизировать объем передаваемых данных и число запросов к веб-серверу.

Использование файлов с JavaScript-кодом большого размера также может негативно сказаться на работе веб-приложения. В ряде случаев вместо этого целесообразно использовать несколько файлов меньшего размера, применять программные средства, такие как LazyLoad JavaScript для динамической загрузки сценариев, разбивая логику веб-приложения на независимые модули.

Следует также ограничить число обработчиков событий в веб-приложении, так как их чрезмерное использование также неминуемо скажется на производительности веб-приложения.

Технология AJAX совершила настоящий прорыв, приблизив веб-приложения к традиционным интерактивным настольным приложениям. При этом разработчики не должны забывать, что AJAX-запрос выполняет полный HTTP-запрос, что эквивалентно по затратам ресурсов перезагрузке все страницы, поэтому количество AJAX-запросов для сложных веб-приложений должно ограничиваться.

Большинство методологий разработки традиционных веб-приложений не воспринимаются для работы, связанной с развитием систем электронной торговли. Автор работы [5] Ю.С. Медведев предлагает, чтобы возникла необходимость в всеобъемлющей структуре развития, в рамках которой можно было бы интегрировать другие подсистемные подходы. Однако любая такая структура должна учитывать стратегические бизнес-факторы системы, эволюционный характер систем, эффективные структуры управления и развитие благоприятной организационной культуры.

Успешная разработка веб-систем и приложений включает в себя несколько интерактивных шагов, которые влияют друг на друга. С. Муругезан и А. Гинидж рекомендуют следующие ключевые шаги для успешной разработки и развертывания веб-приложений [3]:

1. понимать общую функцию и операционную среду системы, включая бизнес-цели и требования, политику организации и политику управления информацией;
2. четко определить заинтересованные стороны, т.е. основных пользователей системы, организацию, которая нуждается в системе, и того, кто финансирует развитие;
3. определить или указать (первоначальные) функциональные, технические и нетехнические требования заинтересованных сторон и всей системы. Кроме того, эти требования не могут оставаться прежними; скорее, они неизбежно будут развиваться со временем во время разработки системы;
4. разработка общей системной архитектуры веб-системы, отвечающей техническим и нетехническим требованиям;
5. определить подпроекты или подпроцессы для реализации архитектуры системы. Если подпроекты слишком сложны для управления, необходимо их разделить еще до того, как они станут набором управляемых задач;
6. разработка и внедрение подпроектов;
7. внедрение эффективных механизмов для управления эволюцией, изменением и обслуживанием веб-приложения. По мере развития системы повторить весь процесс или некоторые его части, если это необходимо;
8. рассматривать нетехнические вопросы, такие как пересмотренные бизнес-процессы, организационная и управленческая политика, культурные и социальные аспекты;
9. измерить производительность системы, проанализировать использование веб-приложения из веб-журналов и просмотреть отзывы пользователей и предложения;
10. обновить систему.

А.Дж. Мадхури рассматривает создание одностраничных веб-приложений с использованием упомянутого фреймворка AngularJS. Согласно мнению автора, AngularJS помогает создавать веб-приложения, основанные на HTML, CSS, JavaScript, в которых AngularJS предоставляет возможности MVC для веб-приложения и, следовательно, делает его более модульным и легким в разработке, обслуживании и тестировании.

AngularJS вводит дополнительные теги, которые называются директивами. Целью директивы является привязка данных к представлению через контроллер. Контроллеры AngularJS написаны на JavaScript, они же и добавляют логическую обработку к представлениям, которые представляют собой не что иное, как HTML-страницы [6].

Согласно отечественным и зарубежным исследованиям, посвященным теме повышения производительности веб-приложения, необходимо минимизировать объем передаваемых данных и число запросов к веб-серверу, применить производительные конструкции кода, кэшировать данные на стороне сервера, использовать многоуровневую архитектуру FrontEnd-BackEnd, а также использовать веб-сервер, построенный по FSM (Finite State Machine).

Для того чтобы успешно создавать сложные веб-приложения для взаимодействия поставщиков и потребителей, веб-разработчикам необходимо принять дисциплинированный процесс разработки и надежную методологию, использовать лучшие инструменты для разработки и следовать ряду хороших рекомендаций.

Следует отметить, что рассмотренные методы в отдельности могут не привести к значительному повышению производительности веб-приложения, однако проведение комплекса предлагаемых мероприятий может значительно увеличить производительность веб-приложения, что особенно актуально для приложений с большим трафиком.

Литература

1. Токарчук А.М. Повышение эффективности методов и алгоритмов разработки, взаимодействия и хранения веб-приложений: автореф. дис. канд. техн. наук. – М.: ФГБОУ ВПО МИИТ, 2012. – 163 с.
2. Сенина А.А., Тузовский А.Ф. Обзор основных современных технологий разработки веб-приложений // Томский политехнический университет. – 2015. – С. 233–235.
3. Murugesan S., Ginige A. Web engineering: introduction and perspectives // International Conference on Web Engineering (ICWE). – 2005. – P. 30.
4. Золотарев Е.В. Создание некоммерческого веб-приложения с использованием гибких методологий разработки // Информатизация инженерного образования. – 2013. – № 1. – С. 80–83.
5. Медведев Ю.С. Проектирование интерактивных веб-приложений // Современные проблемы науки и образования [Электронный ресурс]. – Режим доступа: <http://www.science-education.ru/ru/article/view?id=17486> (дата обращения: 06.12.2017).
6. Madhuri A.J., Sawant B.R., Deshmukh A. Single Page Application using AngularJS // International Journal of Computer Science and Information Technologies. – 2015. – V. 6(3). – P. 2876–2879.

**Сенин Артем Дмитриевич**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4108Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: artemiquss@yandex.ru

**Государев Илья Борисович**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: goss@corp.ifmo.ru

УДК 004.021**АНАЛИЗ ЗАРУБЕЖНЫХ И ОТЕЧЕСТВЕННЫХ ИСТОЧНИКОВ ПО ТЕМЕ
«МОДЕЛИ РЕАКТИВНОГО ПРОГРАММИРОВАНИЯ В ВЕБЕ
И ИХ ИМПЛЕМЕНТАЦИЯ»****Сенин А.Д., 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 исследован вопрос применения модели реактивного программирования в веб. Рассмотрены основные составляющие и примеры шаблонов проектирования, обеспечивающих реализацию реактивного программирования в вебе.

Ключевые слова: модели программирования, реактивное программирование, MVC, 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 современном мире скорость разработки сайтов и веб-приложений имеет большое значение, так как с каждым днем растет спрос на данные виды услуг, что требует в короткие сроки выполнять поставленные задачи. При этом также согласно исследованиям [1–3] важна возможность поддержки и структуризации программного кода.

Существующие модели программирования имели ряд недостатков при решении современных задач по работе с обновляющимися данными. Согласно исследованию Е.И. Маркина [4] к таким недостаткам можно отнести нижеперечисленные показатели.

1. Возможность выхода программы из строя из-за изменения данных в любой момент времени работы программы.
2. Изменение состояний программы при помощи функций обратного вызова, порождающие большие уровни вложенности, что приводит к усложнению восприятия кода программистом и затруднением отладки программы.

3. Дублирование одинаковых данных, вызванное необходимостью использования последних в разных формах.

Реактивный подход к программированию позволяет избежать указанных выше проблем. В основе реактивного подхода лежат потоки данных [5].

Эти потоки асинхронны, и представляют собой последовательность событий, упорядоченных по времени, общая схема которых представлена на рис. 1. Поток может отдавать в качестве ответа три типа данных: значение (определенного типа), ошибку или сигнал успешного завершения. После перехвата этих событий асинхронно, вызывается необходимая функция в зависимости от возвращенного значения. Для этого требуется три функции: функция, вызываемая после получения значения, функция обработки ошибки, функция выполняющая код для обработки сигнала завершения.

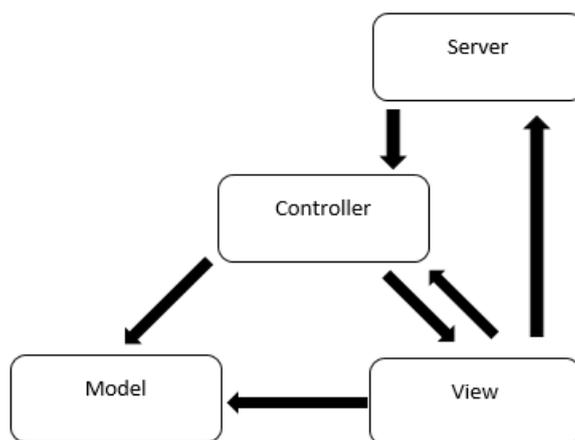


Рис. 1. Логика работы приложения, построенного на основе MVC

Слежение за потоком называется «подпиской», или прослушиванием потока. Функции, которые объявляются в коде, называются «наблюдатели». Таким образом, поток – объект наблюдений. Такой алгоритм решения задачи соответствует шаблону проектирования «наблюдатель».

Паттерн «наблюдатель» относится к классу поведенческих шаблонов и позволяет классу-наблюдателю получать оповещения от класса-наблюдаемого об изменении его состояния, реализуя, таким образом, схему взаимодействия «издатель-подписчик». В качестве наблюдателя при реализации шаблона MVC выступает «вид» или различные «виды», а наблюдаемым является модель. Модель оповещает виды об изменении своего состояния, после чего виды извлекают необходимые данные и динамически генерируют веб-страницу.

Исходя из описанного выше принципа работы реактивной модели программирования и обобщая практический опыт программистов [6], участвовавших в обсуждении данной темы, можно сказать, что в случае с применением реактивного программирования при разработке сайта, реактивный подход обладает следующими преимуществами по сравнению с классическими моделями программирования:

1. отзывчивость интерфейса;
2. масштабируемость;
3. отказоустойчивость.

Применение паттерна «наблюдатель» также обосновано в этой системе и тем, что помимо потока событий имеется пользовательский интерфейс, посредством которого происходит взаимодействие с программой. Анализ исследования [2] показал, что для проектирования архитектуры веб-приложений наиболее востребованным на сегодняшний день является MVC-паттерн, представляющий собой разделение программной системы на несколько компонентов, называемых «модель» (model), «вид» (view) и «контроллер»

(controller), обеспечивающих четкое разделение бизнес-логики приложения и пользовательского интерфейса. Реализацию между видом и моделью возможно различными способами, в том числе и при помощи паттерна «наблюдатель».

Изучением проблемы контроля состояние приложения и его визуального отображения занимался также В.Л. Чугреев [3]. Данная проблема рассматривается В.Л. Чугреевым в контексте применения шаблона проектирования MVC, что расшифровывается как model, view, controller.

Как пишет автор «его основная идея в том, чтобы отделить модели данных, их отображение и операции взаимодействия с пользователем» [3, С. 3]. Положительная сторона такой организации кода заключается в том, что она позволяет распределить код по уровням, каждый из которых определяет сферу ответственности и позволяет грамотно упорядочить архитектуру приложения при проектировании. Такие программные компоненты менее зависимы друг от друга, и это облегчает их дальнейшее модифицирование, благодаря чему доработка и развитие такой системы становится проще. Общая схема работы приложения, построенного по архитектуре MVC, представлена выше на рис. 1.

В общем случае такая система работает по следующему принципу. Обработкой запросов пользователя занимается контроллер, который после получения запроса обрабатывает его и запрашивает данные у соответствующих моделей, после получения данных от модели, агрегирует их с другими данными и передает данные в представление.

Согласно исследованию А.Н. Моисеева [7], организация MVC берет начало в таких паттернах проектирования как «посредник» (mediator) и «наблюдатель» (observer).

Паттерн посредник применяется для реализации контроллера интерфейса, он предполагает полную изоляцию классов (в данном случае элементов интерфейса view) друг от друга, в отличие от типичной структуры связей между классами представления представлены на рис. 2, а. В данном случае при такой организации все их взаимодействие осуществляется через специально выделенный объект-посредник, который и является контроллером интерфейса пользователя на рис. 2, б. Класс контроллера инкапсулирует все управление интерфейсом и начальную обработку все событий, что повышает уровень управляемости системы.

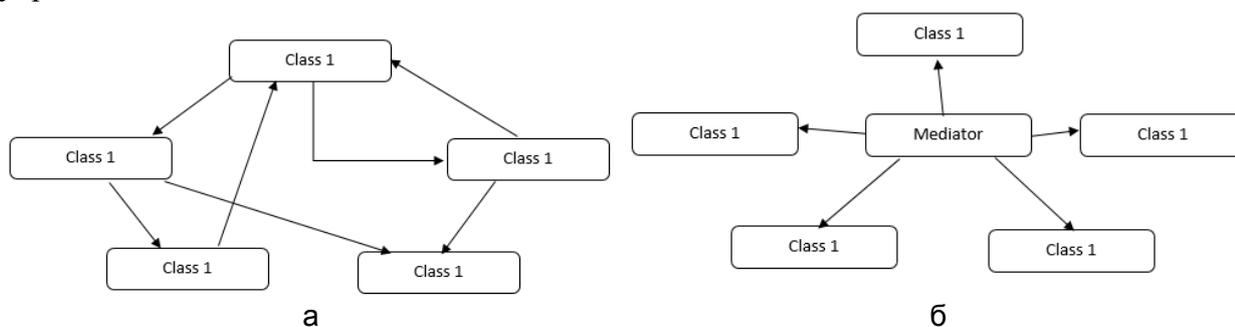


Рис. 2. Типичная структура классов-представлений (а); применение паттерна «посредник» для связи представлений (б)

Как пишет А.Н. Моисеев «недостатком данного подхода является строгая привязка интерфейсных объектов к классу контроллера – к классам интерфейсных элементов» [7, С. 156]. Однако такой недостаток может быть устранен при помощи выделения абстракций контроллера-посредника и интерфейсного элемента, и применения паттерна «наблюдатель», который реализует взаимодействие конкретных классов [8].

Применение реактивного программирования в современном сайтостроении в первую очередь особенно востребовано при создании одностраничных веб-приложений. Это связано с тем что размер интерфейса сильно ограничен по сравнению с разработкой веб-страниц для настольных компьютеров, что требует возможности уместить под различными скрытыми по умолчанию элементами управления дополнительный функционал приложения, а также

повышения производительности приложения благодаря уменьшению количества и увеличению качества производимых операций с поступающими на устройство данными.

Применение реактивного программирования в современном сайтостроении в первую очередь востребовано при создании одностраничных веб-приложений. Это связано с тем, что размер интерфейса сильно ограничен по сравнению с разработкой веб-страниц для настольных компьютеров, что требует возможности уместить под различными скрытыми по умолчанию элементами управления дополнительный функционал приложения, а также повышения производительности приложения благодаря уменьшению количества и качественному увеличению качества производимых операций с поступающими на устройство данными.

Реактивное программирование, будучи примененным в сфере веб, решает очень важные задачи, связанные с реализацией как пользовательских интерфейсов, так и работы с данными получаемыми со стороны сервера, а также работы с состоянием веб-приложения, обеспечивая удобную с точки зрения написания кода программы и последующей поддержки продукта, и с точки зрения удобства пользователя данного продукта реализацию веб-приложения.

Литература

1. Елисеев Б.П. Программный паттерн проектирования архитектурного каркаса Model-View-Controller при разработке веб-приложений систем мониторинга спецтранспорта аэропорта // Научный вестник МГТУ ГА. – 2015. – № 220(10). – С. 137–142.
2. Лясин Д.Н., Симонова О.Н. Реализация архитектурного шаблона MVC с использованием шаблона проектирования «Наблюдатель» на языке PHP // Молодой ученый. – 2014. – № 6. – С. 108–111.
3. Чугреев В.Л. Особенности реализации MVC-архитектуры в веб-приложениях // Молодой ученый. – 2015. – № 7. – С. 66–71.
4. Маркин Е.И., Рябова К.М. Реактивное программирование // Вестник современных исследований. – 2016. – № 2-1(2). – С. 112–117.
5. Маркин Е.И., Рябова К.М., Пименов А.Н., Левина Н.А., Воронцов А.А. Технология реактивного программирования // Сб. статей победителей IV международной НПК «Инновационные научные исследования: теория, методология, практика». – 2016. – С. 44–47.
6. Какие задачи решает реактивное программирование? [Электронный ресурс]. – Режим доступа: <https://toster.ru/q/290750> (дата обращения: 06.12.2017).
7. Моисеев А.Н. Контроллер интерфейса пользователя с повышенной степенью повторной используемости // Вестник Том. гос. ун-та. – 2006. – № 293. – С. 156–157.
8. Роганов Е.А. Основы информатики и программирования. – М.: МГИУ, 2001. – 315 с.

**Сенчило Марина Сергеевна**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4106Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: ma-ri-na__95@mail.ru

**Шуклин Дмитрий Анатольевич**

Год рождения: 1979

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: do@limtu.ru

УДК 004**ИССЛЕДОВАНИЕ ТЕХНОЛОГИЙ АЛГОРИТМИЧЕСКОГО ДИЗАЙНА
ИНТЕРНЕТ-ПРОЕКТОВ****Сенчило М.С.****Научный руководитель – к.педагог.н., доцент Шуклин Д.А.**

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

В данной работе приведен обзор технологий алгоритмического дизайна, определены особенности алгоритмического дизайна интернет-проектов и рассмотрены примеры использования алгоритмического дизайна в проектах.

Ключевые слова: алгоритмический дизайн, разработка, технологии, дизайн, цифровое проектирование.

В современном мире алгоритмический дизайн имеет большое значение, так как предлагает нетрадиционный способ концептуализации и работы в дизайне. Алгоритмические системы тесно связаны с современным проектированием. Интеграция в процессе проектирования позволяет разработать новые дизайнерские решения, которые трудно или невозможно достичь посредством других методов. Алгоритмические методы используют принципы базы данных, генерирование сложных форм и шаблонов из простых технических характеристик. Эволюционные системы могут использоваться в сочетании с эстетическим выбором для разработки дизайнерских решений под руководством дизайнера.

На сегодняшний день алгоритмические средства проектирования, которые производят оптимальные формы для продуктов и зданий без вмешательства человека, преобразуют как физический мир, так и роль дизайнера. Программное обеспечение может автоматически упростить работу самолетов, создание зданий, разработку программ – все это более комфортно с дизайнером, действующим как «куратор», а не как исполнитель всех решений.

Новая технология использует алгоритмы для генерации всех возможных перестановок дизайнерского решения. Дизайнер просто вводит набор параметров, а затем выбирает лучший результат, созданный программным обеспечением (ПО) [1].

Преимущества алгоритмического дизайна:

- изучает более широкий спектр вариантов дизайна. В то время, когда дизайнер может создать одну идею, компьютер может сгенерировать тысячи, а также данные, чтобы доказать, какие проекты лучше всего работают;
- создает новые проекты, алгоритмический дизайн позволяет создавать оптимизированные сложные формы, некоторые из этих форм невозможно сделать традиционными методами производства;
- оптимизирует материалы, методы производства и стоимость, задаются цели, параметры и ПО создаст высокопроизводительные варианты дизайна на основе этих ограничений, программное обеспечение разрешает противоречивые ограничения дизайна, поэтому разработчик, дизайнер может сосредоточиться на инновациях [2].

Работа программы, используемой алгоритмический дизайн, начинается с конструктора, определяющего область ограничения/проектирования, точек подключения и препятствий, а также множество сдерживающих параметров. Пример первых трех можно увидеть, где эти параметры определены для дизайна мотоциклетного маятника. Оптимизация основана на желаемых материалах, технологиях производства (способе производства), температурной толерантности, стоимости и прочности части, и ее способности выдерживать указанные силы.

Обычно дизайнеру приходится тратить несколько часов, пытаясь подсчитать, будет ли их предлагаемый дизайн соответствовать этим ограничениям. Напротив, это делается по умолчанию с использованием алгоритмического дизайна, и все результирующие конструкции будут соответствовать определенным ограничениям и часто превышают их. Благодаря этой эффективной автоматизации задач с помощью ПО у дизайнера есть возможность поиграть с эстетикой этой части и инвестировать больше времени в изучение творческих и эффективных проектов [1].

Таким образом, весь процесс алгоритмического дизайна можно разбить на следующие правила: в первую очередь формируется идея от идеи, программа обращается к правилам, алгоритмам. На этом этапе происходит формализация и формируются стартовые параметры, создается код, после идет исполнение кода компьютером, формируется результат. На всех стадиях дизайнер может внести свои изменения: изменить правила, изменить исходный код и параметры, оценить результат. В случае если дизайнер не удовлетворен результатом, он может начать процесс заново [3].

Алгоритмический дизайн получил широкое применение в таких сферах как: промышленный дизайн, веб-дизайн, архитектура, графический дизайн, визуализация данных, медицина, искусство и индустрия развлечений. Так, например, в сфере промышленного дизайна алгоритмический дизайн используют для снижения веса отдельных элементов, синтеза формы, создания медицинских имплантов.

В графическом дизайне и визуализации данных при помощи алгоритмического дизайна создаются фирменный стиль, дизайн плакатов и упаковок, визуализация данных и инфографика. В веб-дизайне программа сама способна распознавать макет, изображение или скриншот сайта и генерирует готовую интерактивную страницу с кодом, оформлением и графическими элементами. Также программа позволяет сделать редизайн веб-страниц, т.е. заказчик получает новый, оптимизированный сайт, сгенерированный на основе материалов со старой версии сайта и программа автоматически прикрепляет ее к существующему домену [4].

Алгоритмический дизайн, который имитирует развитие организмов в естественном мире, стал главной темой в университете Autodesk – трехдневной технологической конференции, прошедшей в Лас-Вегасе в ноябре прошлого года. «Алгоритмический дизайн – это отступление от традиционного дизайна» – сказал Джефф Ковальски, главный технический директор программного обеспечения Autodesk. «Но эти технологии не представляют угрозы, они больше похожи на сверхдержавы».

Метод цифрового проектирования основывается на облачных вычислениях для создания множества вариантов на основе набора фиксированных параметров. Такой метод часто создает жидкие, решетчатые формы, поскольку ПО имитирует способ создания наиболее эффективных структур. Результат можно увидеть в спортивном бренде New Balance, выпущенном в 2015 году. Новая технология использует алгоритмы для генерации всех возможных перестановок дизайнерского решения, в результате чего получаются такие продукты, как 3D-печатные подошвы для тренажеров [5].

Дизайнер Фрэнсис Битонти недавно использовал генеративный дизайн, чтобы уменьшить размер и вес ортопедических брекетов, которые носят люди страдающие заболеванием сколиоз. Сам дизайнер о своей разработке говорит: «Генерирующий дизайн имитирует способ эволюции организмов в естественном мире, создавая самые легкие структуры». Ортопедический брекеты разработанный дизайнером Фрэнсисом Битонти.

В начале этого года Autodesk запустили собственный генераторный дизайн Dreamcatcher. Он интегрирован в некоторые из современных программ 3D-дизайна компании, таких как Fusion 360 и Inventor. Autodesk Dreamcatcher – это облачное ПО для 3D-дизайна, которое предоставляет разработчикам возможность программно оптимизировать 3D-дизайн [3].

Dreamcatcher был разработан, чтобы иметь возможность обрабатывать более широкий спектр параметров, чем предыдущие генеративные средства проектирования, и уже используется для настройки интерьеров и даже проектирования целых зданий.

Алгоритмический дизайн позволяет сэкономить время на процессе проектирования, сохранить материал, создав наиболее эффективные структуры и сэкономить средства, разработав наиболее рентабельный способ их производства. Но это не без потенциальных рисков. Существует риск того, что работа дизайнера будет упрощена, исключив значительную часть творческого процесса из своей работы – необходимо будет просто выбрать оптимальный набор параметров. Ведущие компании стремятся обеспечить, чтобы алгоритмический дизайн позволял дизайнерам продвигать формы и функциональность продуктов еще больше, чем когда-либо прежде, тогда как некоторые специалисты отмечают, что роль дизайнера, безусловно, изменится по мере того, как техника станет более широко принятой.

На основе вышеперечисленного можно отметить, что алгоритмический дизайн позволяет достигнуть наилучшего, оптимального результата при этом минимизируя затраты на средства, материал, время.

Литература

1. Легалов А.И. Языковые средства систем программирования, ориентированные на создание переносимых, эволюционно расширяемых параллельных программ. – Красноярск, 2014. – С. 112–236.
2. Лукичев Р.В. Фактор случайности как принцип генеративного искусства [Электронный ресурс]. – Режим доступа: <https://sibac.info/conf/philolog/xxx/35101>, своб.
3. Shiffman D. Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction. – Изд-во: Morgan Kaufmann, 2013. – 450 p.
4. Роджерс Д. Алгоритмические основы машинной графики. – М.: Мир, 1989. – 511 с.
5. Reas C., Fry B. & Maeda J. Processing: A Programming Handbook for Visual Designers and Artists. – The MIT Press, 2013. – 736 p.



Сергейчук Николай Андреевич

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: nicknixer@gmail.com



Готская Ирина Борисовна

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, д.педагог.н., профессор

e-mail: iringot@mail.ru

УДК 004

**СИСТЕМЫ ВЕБ-АНАЛИТИКИ: СФЕРЫ ПРИМЕНЕНИЯ И ДАЛЬНЕЙШИЕ
ПЕРСПЕКТИВЫ**

Сергейчук Н.А.

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 проведен обзор сфер применения современных систем веб-аналитики. Приведены способы использования систем веб-аналитики в различных сферах применения. Сформулированы выводы о дальнейших перспективах на основании исследований, ведущихся в этой области.

Ключевые слова: информационные системы, веб, аналитика, Интернет, ИТ, IT, Google Analytics, Yandex.Metrica, Яндекс.Метрика, Гугл Аналитикс.

В настоящее время использование сети Интернет приобрело массовый характер. Для большинства пользователей сеть Интернет не только обеспечивает быстрый доступ к необходимой информации, но и является основным источником новостей, средством для общения. В бизнесе также широко используют сеть Интернет, многие компании имеют свои собственные сайты, которые позволяют решать определенные бизнес-задачи.

Сфера бизнеса, где все финансовые транзакции и торговля осуществляются с помощью сети Интернет, называется электронной коммерцией. Эффективность такого бизнеса зависит от посещаемости и аудитории интернет-ресурса. Торговой интернет-площадке очень важно иметь большую посещаемость, от этого зависит количество продаж.

Одной из сфер, где тоже широко используются системы веб-аналитики, является туризм. В работе С.А. Демьянова [1] о практическом использовании инструментария Google Analytics в туристическом бизнесе отмечается, что современный туристический бизнес невозможно представить без Интернета, потому что туристические порталы, сайты туристических фирм, туристические онлайн-гипермаркеты, глобальные системы бронирования и резервирования, и иные веб-ресурсы стали неотъемлемой частью этого бизнеса. С активным развитием веб-ресурсов возникают вопросы, связанные с возможностью проведения мониторинга посещаемости сайтов, а также оценкой экономической эффективности рекламных интернет-кампаний и оптимизацией структуры и содержимого сайта. Системы веб-аналитики как раз призваны решать задачи, связанные с мониторингом посещаемости сайтов.

В блогосфере и социальных медиа системы веб-аналитики являются необходимыми инструментами для анализа аудитории. Авторам тематического контента важно знать, какой контент пользуется большей популярностью, чтобы вести проект в направлении, которое позволит ему развиваться эффективнее и быть интересней.

Системы веб-аналитики находят применение и в научной сфере. Популяризация научных достижений является одной из важнейших задач любой организации, выполняющих научно-исследовательские работы. Это подтверждается проектом Министерства образования и науки РФ, в котором заявляется о необходимости учета не только количества упоминаний о такой организации в СМИ, но и посещаемости официальных сайтов в сети Интернет. В работах российских ученых [2, 3] отмечается, что современное развитие информационных технологий и сети Интернет, в частности, существенным образом повлияли на деятельность научных и образовательных организаций; что отдельные исследователи и организации в целом получили огромные возможности по продвижению результатов своих трудов и взаимодействию с российскими и зарубежными коллегами посредством средств электронной коммуникации, поэтому требуется постоянный контроль и оценка результатов выполняемой работы этих организаций с целью принятия своевременных управленческих решений, для корректировки направлений дальнейшего развития. Одной из задач, направленных на достижение указанной цели, является оценка популярности веб-сайтов научной организации с помощью анализа статистики их посещения.

Еще одной сферой, где широко применяются системы веб-аналитики, является интернет-маркетинг. Задача любого бизнеса – получение прибыли, это относится и к интернет-магазинам, для которых выполнение этой задачи напрямую связано с доходом от заказов, которые, в свою очередь, зависят от выбора покупателями конкретных товаров. В связи с этим важно уметь продвигать нужные позиции, так как доход от продаж товаров даже одного ценового сегмента и схожих покупательских свойств отличается. В работе [4] о способе ранжирования, повышающем доход с заказа в интернет-магазине В.Л. Флак обращает внимание на то, что при продвижении интернет-площадки важно знать, какие источники приводят больше клиентов, чтобы успешнее распорядиться рекламной компанией. Результаты анализа научных трудов зарубежных и российских ученых и специалистов позволили сделать вывод, что системы веб-аналитики нашли широкое применение в сфере онлайн-образования. Данные системы используются для аналитики интернет-площадок с различными образовательными курсами и учебной литературой, помогают определить более интересную информацию для посетителей и задавать более правильный вектор при выборе тематики литературы и курсов, на которых стоит ставить приоритет при наполнении площадки контентом, что может способствовать лучшему развитию площадки, так как на сайте будет появляться только интересный контент.

Также важно отметить, что системы веб-аналитики нашли применение в сфере онлайн-правительства в некоторых странах. На российском портале предоставления государственных услуг (<https://www.gosuslugi.ru/>) также могут применяться такие системы.

Анализируя сферы применения систем веб-аналитики, можно сделать вывод, что такие системы используются везде, где возникает задача создания, ведения и продвижения интернет-ресурса, а также должна быть предоставлена возможность получения подробной информации об аудитории, что может способствовать выявлению более продуктивных средств продвижения.

В настоящее время ведутся исследования по применению систем веб-аналитики в сфере культуры, вопросом использования системы на сайтах виртуальных тематических музеев занимались такие зарубежные ученые как Anggai S., Blekanov I.S. и Sergeev S.L [5]. Очевидно, в дальнейшей перспективе системы веб-аналитики будут развиваться и находить применение в новых сферах деятельности человека (не только профессиональной) и, возможно, выйдут за рамки веб-пространства, занимая нишу Интернета вещей и виртуальной реальности.

Литература

1. Демьянов С.А. Инструментарий Google Analytics и его практическое использование в туристическом бизнесе // Туризм и гостеприимство. – 2016. – № 1. – С. 44–50.
2. Кабакова Е.А., Усков В.С. Веб-сайт научно-исследовательского учреждения: наполнение, посетители, развитие [Электронный ресурс]. – Режим доступа: <http://vtr.isert-ran.ru/article/1396/full> (дата обращения: 12.11.2017).
3. Скородумов П.В., Холодев А.Ю. Анализ подходов и инструментальных средств анализа статистики посещения веб-сайта научной организации // Вопросы территориального развития. – 2015. – № 9(29). – С. 6.
4. Флакс В.Л. Способ ранжирования товаров, повышающий конверсию и доход с заказа в интернет-магазине // Наука и прогресс транспорта. Вестник Днепропетровского национального университета железнодорожного транспорта. – 2012. – № 42. – С. 197–200.
5. Anggai S., Blekanov I.S., Sergeev S.L. Design muntoi WEB-based framework and search engine analytics for thematic virtual museums // Proceedings of the 4th international conference on interactive digital media. – 2015. – P. 51–63.

**Сивинский Станислав Андреевич**

Год рождения: 1992

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: stanislav_sivin@mail.ru

**Готская Ирина Борисовна**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, д.педагог.н., профессор

e-mail: iringot@mail.ru

УДК 004.9**К ВОПРОСУ РАСШИРЕНИЯ ВОЗМОЖНОСТЕЙ СИСТЕМ ДИСТАНЦИОННОГО
ОБУЧЕНИЯ MOODLE ДЛЯ ФОРМИРОВАНИЯ РЕЗУЛЬТАТА ТЕСТИРОВАНИЯ
ПО РАЗЛИЧНЫМ КРИТЕРИЯМ****Сивинский С.А.****Научный руководитель – д.педагог.н., профессор Готская И.Б.**

Работа выполнена в рамках темы НИР № 643202 «Исследование методов и алгоритмов формирования облачных информационных систем».

В работе рассмотрены проблемы проведения тестирования с оценкой результатов по различным критериям в рамках применения системы дистанционного обучения Moodle. Рассмотрены примеры возможных решений за счет установки специальных программных модулей (плагинов), позволяющих увеличить базовые возможности системы дистанционного обучения Moodle. А также предложен вариант решения поставленной проблемы за счет разработки и внедрения авторского плагина «Профориентационное тестирование».

Ключевые слова: Moodle, профориентационное тестирование, плагин, разработка плагина, дифференциально-диагностический опросник.

Информатизация жизнедеятельности общества позволила внедрить новые технологии в образовательный процесс школы и вуза. Появление на рынке образовательных технологий систем дистанционного обучения (СДО) позволило разработать новые технологии обучения (дистанционное обучение, адаптивное обучение, смешанное обучение).

Эффективность образовательного процесса в первую очередь зависит от заинтересованности обучающегося результатом обучения и от собственных целевых установок. С точки зрения психологии обучающийся может быть предрасположен в большей или меньшей степени к той или иной отрасли жизнедеятельности. Осознанный выбор профессии является ключом к успешному становлению личности и раскрытию всего потенциала заложенного в ней. Процесс профессиональной ориентации – это система научно-обоснованных методов, направленных на подготовку учащегося к осознанному выбору будущей образовательной программы [1].

Одним из вариантов профессиональной ориентации является метод тестирования по системе дифференциально-диагностического опросника (ДДО) Е.В. Климова [2]. ДДО

предназначен на определение предрасположенности личности к той или иной сфере деятельности человека на основе ценностных ориентаций.

В образовательной практике широко распространено применение СДО (Moodle, OpenEdx, CanvasLMS и др.). Среди всех систем наиболее распространенной и удобной с точки зрения конфигурирования и расширения функционала является Moodle.

Система Moodle позволяет в кратчайшие сроки развернуть и настроить «классический процесс» дистанционного обучения. Система включает несколько основных модулей: лекций, тестирования, заданий, журнал оценок и личный кабинет пользователя. Функционал системы может быть расширен за счет установки дополнительных модулей.

СДО Moodle может быть успешно применена для проведения психологического тестирования, где накопление баллов происходит по одному критерию. К сожалению, базовый модуль тестирования Moodle не может быть применен для организации процесса тестирования по методу ДДО Е.В. Климова.

Для организации тестирования по методу Е.В. Климова необходимо формирование результата по различным критериям. На рис. 1, а, показана блок-схема классического тестирования. В классическом процессе тестирования критерий формирования результата один и помогает оценить уровень ответов обучающегося от 0% до 100% в зависимости от количества полученных верных ответов и суммарного веса всего теста.

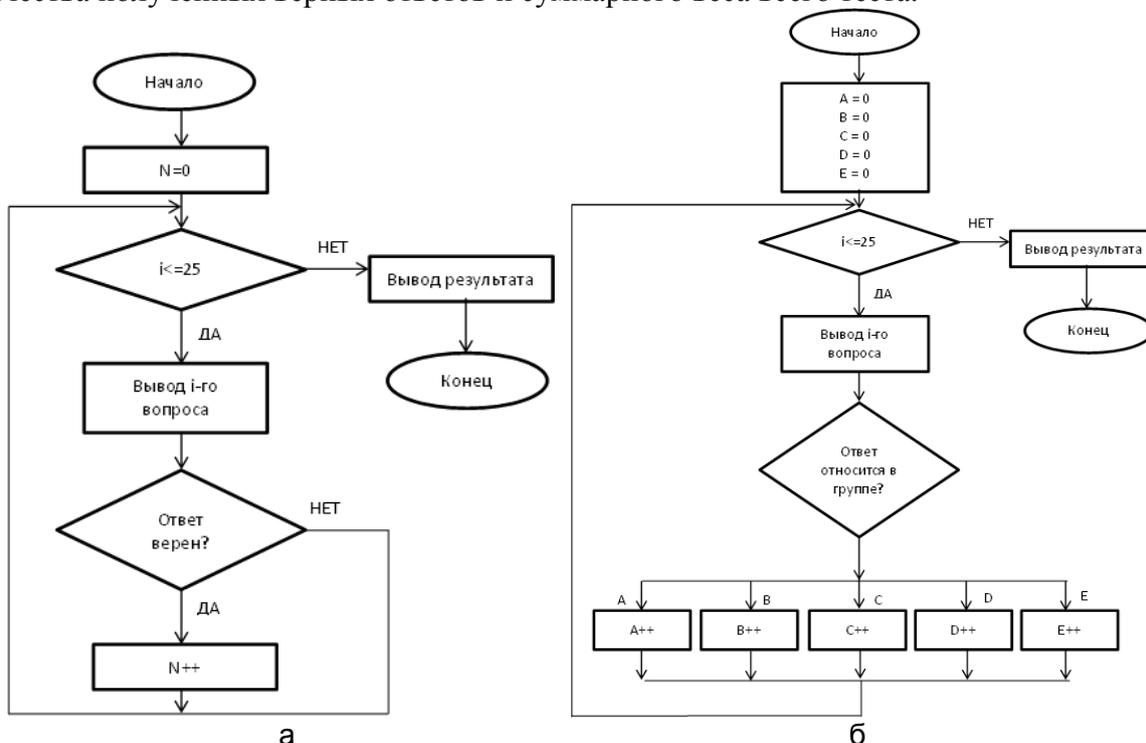


Рис. 1. Классическая система тестирования (а); система тестирования по методу Е.В. Климова (б)

Ключевое отличие методологии тестирования ДДО показано на рис. 1, б. В отличие от классического тестирования ДДО предполагает наличие нескольких критериев оценки для формирования результата тестирования. Так, на блок-схеме представлено 5 критериев, которые связаны с ответами из списка вопросов. В данном тестировании нет верных или неверных ответов. Если обучаемый выбирает ответ «А», то в зависимости от схемы теста разработанной автором, ответ может относиться к одной из 5-ти предложенных категорий.

Для решения поставленной проблемы автором был проведен анализ существующих дополнительных модулей (плагинов) системы Moodle. Среди доступных решений можно выделить плагин Multitrack Test [3].

Данный модуль позволяет создавать несколько критериев для оценки результатов тестирования. Но при создании списка вопросов все созданные критерии присутствуют в качестве ответов, не позволяя создавать связь между отдельным вариантом ответа и доступными критериями.

Для решения проблемы организации профессиональной ориентации на базе СДО Moodle был разработан собственный модуль «Профориентационное тестирование». В соответствии с рис. 2 показан интерфейс модуля.

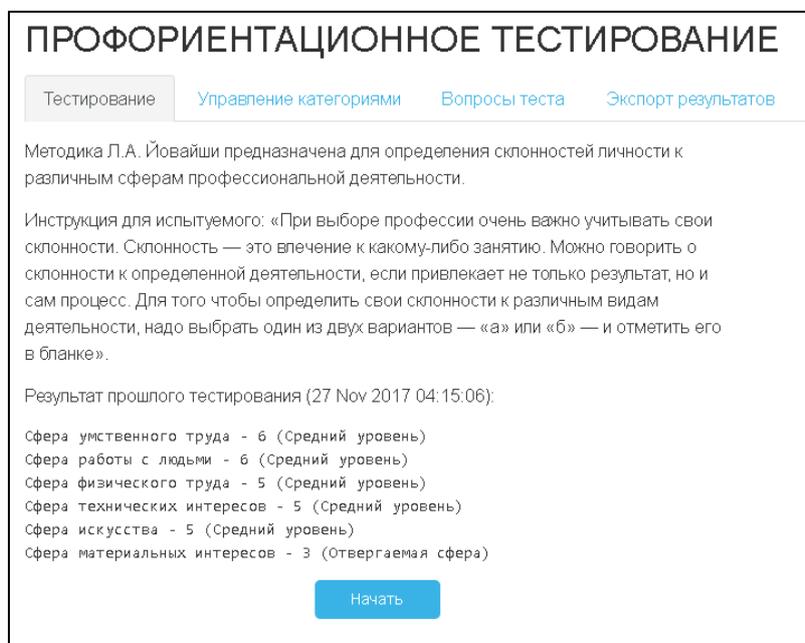


Рис. 2. Интерфейс модуля «Профориентационное тестирование»

Модуль «Профориентационное тестирование» предлагает возможность не только создания нескольких критериев, но и связи вариантов ответов с критериями.

В разделе «Управление категориями» автор теста создает список категорий (критериев). Этап создания вопросов в разделе «Вопросы теста» невозможен до тех пор, пока не будут созданы категории.

Для организации удобного процесса последующей работы с результатами тестирования была создана возможность экспорта результатов тестирования в разделе «Экспорт результатов» (рис. 3).

ПРОФОРИЕНТАЦИОННОЕ ТЕСТИРОВАНИЕ

Тестирование Управление категориями Вопросы теста Экспорт результатов

Выберите поток... Выберите группу... Экспорт полностью Экспорт выделенное

Количество строк (в таблице): 5

Назад 1 с. Вперед

<input type="checkbox"/>	Имя пользователя	Логин	Результат	Дата
<input type="checkbox"/>	Станислав Сивинский	admin	Сфера умственного труда - 6 (Средний уровень) Сфера работы с людьми - 6 (Средний уровень) Сфера физического труда - 5 (Средний уровень) Сфера технических интересов - 5 (Средний уровень) Сфера искусства - 5 (Средний уровень) Сфера материальных интересов - 3 (Отвергаемая сфера)	2017-11-27 04:15:06
<input type="checkbox"/>	Станислав Сивинский	admin	Сфера работы с людьми - 1 (Отвергаемая сфера) Сфера искусства - 1 (Отвергаемая сфера)	2017-11-17 23:59:54
<input type="checkbox"/>	Ольга Гордиевская	student	Сфера искусства - 9 (Предпочитаемая сфера) Сфера материальных интересов - 7 (Предпочитаемая сфера) Сфера технических интересов - 6 (Средний уровень) Сфера физического труда - 4 (Средний уровень) Сфера работы с людьми - 3 (Отвергаемая сфера) Сфера умственного труда - 1 (Отвергаемая сфера)	2017-09-23 21:52:32

Рис. 3. Интерфейс раздела «Экспорт результатов»

Интерфейс модуля предполагает разделение ролей для студента и преподавателя, таким образом, вкладки «Управление категориями», «Вопросы теста», «Экспорт результатов» недоступны для студентов.

Экспорт результатов может быть отфильтрован по потоку, группе и осуществляется в табличный процессор MS Excel. Экспортировать можно как полностью все результаты тестирования, так и отдельные записи из таблицы.

Литература

1. Берестнева О.Г., Фисоченко О.Н., Моисеенко А.В., Щербаков Д.О. Разработка профориентационной системы поддержки принятия решения для абитуриентов Национального исследовательского Томского политехнического университета // Интернет-журнал Науковедение. – 2013. – № 4(17). – С. 76.
2. Дифференциально-диагностический опросник Климова [Электронный ресурс]. – Режим доступа: <https://psychologys.ru/oprosnik-klimova/>, своб.
3. MultitrackTest [Электронный ресурс]. – Режим доступа: https://moodle.org/plugins/mod_magtest, своб.

**Слепухин Ярослав Игоревич**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4107Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: yaroslavslepukhin@gmail.com

УДК 004.77**СРЕДСТВА КОНТРОЛЯ И УЛУЧШЕНИЯ КАЧЕСТВА ПРОГРАММНОГО КОДА
НА ЯЗЫКЕ JAVASCRIPT****Слепухин Я.И.****Научный руководитель – к.ф.-м.н. Соловьев И.А.**

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

В работе рассмотрены основные средства мониторинга и поддержания качества кода на языке JavaScript. Дан краткий обзор каждого представленного способа. В заключении обозначены темы для дальнейшего направления исследования.

Ключевые слова: качество кода, JavaScript, разработка, оптимизация.

Введение. Для разработчиков и предпринимателей в сфере IT проблема поддержания высокого качества кода является важной. Это влияет на возможность расширения и поддержки существующего кода, стоимость разработки и качество конечного продукта. Особенно остро встает вопрос для JavaScript-программистов, поскольку эта сфера наиболее динамичная и быстроразвивающаяся, с огромным притоком начинающих специалистов [1].

Практика показывает, что разработчики случайным образом выбирают те или иные методы контроля качества кода исходя из своих привычек. Благодаря исследованиям в данной области, JavaScript-программисты смогут делать выбор в пользу эффективных методов и инструментов.

Необходимо рассмотреть основные средства мониторинга программного кода на языке JavaScript, выработать общие предварительные рекомендации на основании анализа и оценить литературу в исследуемой области. Согласно этому будет выполнена постановка задачи для дальнейшего изучения.

Было выделено пять наиболее распространенных средств для контроля и улучшения качества кода:

1. использование систем контроля версий;
2. инспекции кода;
3. статистические анализаторы JavaScript;
4. стандарты оформления кода;
5. строгая типизация в JavaScript.

Системы контроля версий – это средство для управления изменениями в документах, компьютерных программах, больших сайтах. Такой инструмент позволяет:

- получить доступ к информации: кто сделал изменение, зачем кто-то его сделал, в какое время и саму суть изменения, т.е. код;
- организовать удобную разработку для целой команды программистов. Даже в те моменты, когда разработчики делают конфликтные изменения;
- появляется возможность вернуться к более ранней версии проекта в случае, если кто-то в команде внес изменения, приводящие к ошибкам;

- легко работать над несколькими независимыми функциями или версиями программы, есть возможность быстро переключаться между ними.

Использование Git или другого подобного инструмента сможет повысить удобство работы над проектом, его контроль и тем самым улучшить само качество написания кода. В наши дни такие системы достаточно просты в освоении. Существует хорошо написанная документация и литература: полное руководство по Mercurial от Брайана О'Салливана [2], книга Pro Git [3], написанная Скоттом Чаконом совместно с Беном Страбом и другие.

Инспекция кода – это всего-навсего просмотр кода другим человеком. Главная идея данного метода заключается в том, что человек, непричастный к написанному, может со свежим взглядом найти недостатки, и поэтому предложить более оптимальные решения некоторых проблем. В результате ревизий получается более качественный код, к тому же в процессе инспекций происходит обмен опытом, что повышает общую квалификацию программистов.

Хотя стоимость инспекций код может достигать до 20% от стоимости всего проекта, данные различных исследований, которые приводит Стив Макконнелл [4, С. 493], говорят об очень высокой эффективности: каждый час код ревью предотвращал от 30 до 100 ч работы, которые тратились на тестирование и исправление багов.

Стандарт оформления кода – это конвенция о том, в каком стиле писать программный код. Эти инструкции указывают, какой длины делать отступы, где и как писать комментарии, какой допустимый уровень вложенности для циклов и условных операторов, как именовать функции и так далее.

Стиль программирования призван сделать так, чтобы код выглядел единообразно, будто бы его пишет один программист. Это улучшает читаемость кода, его ясность, позволяет новым членам команды быстро освоиться в написанном и включиться в процесс. К тому же некоторые эксперименты показывают, что следование определенным гайдлайнам может повысить производительность кода [5]. Крупные компании и проекты разрабатывают свои собственные стили. Наиболее известные: Google JavaScript Style Guide, который учитывает нововведения ECMAScript 6, JavaScript стайлгайд от компании Airbnb, стили проекта Node.js и другие.

Статические анализаторы кода помогают отслеживать соблюдение определенных стилей, находить потенциальные проблемы (антипаттерны), выявлять ошибки синтаксиса и следить за следованием спецификации. Данная проверка кода не подразумевает его исполнения.

Стефанов в книге «JavaScript Шаблоны» упоминает один из популярных анализаторов – JSLint. Он пишет следующее: «Этот инструмент «будет задевать ваше самолюбие», как предупреждает его создатель, но только на первых порах. Благодаря JSLint вы будете быстро учиться на собственных ошибках и постепенно выработаете в себе основные привычки профессионального программиста на JavaScript. Отсутствие предупреждений после проверки с помощью JSLint поднимет вашу уверенность в программном коде, так как вы будете знать, что второпях не допустили элементарных оплошностей или синтаксических ошибок» [6].

Строгая типизация как один из методов улучшения качества кода. JavaScript является языком с динамической типизацией, т.е. типы данных определяются в момент выполнения программы. Соответственно, строгая типизация для языка означает указание типа переменной в процессе написания кода. Проблема в том, что по данным исследований ошибки несоответствия типов встречаются довольно часто, и большинство разработчиков оценивают их как критические [7]. В этой связи возникла потребность в инструментах, добавляющих статическую типизацию в JavaScript, чтобы повысить качество исходного кода. Крупные компании начали выпускать свои средства для обеспечения строгости языка:

- Google Web Toolkit от Google;
- TypeScript от Microsoft;
- Flow от Facebook.

Но оценки эффективности представленных средств пока еще сильно разнятся. Однозначных показателей нет. Дуглас Крокфорд пишет: «Я думаю, что слабая типизация в JavaScript является одной из лучших функций, а проверка типов переоценивается. TypeScript добавляет достоинства, но это не та цена, которую я готов заплатить» [8].

Заключение. Проведена краткая оценка каждого представленного способа. Выявлено, что на сегодняшний день существуют как общепринятые методы, которые прочно вошли в канон современной JavaScript разработки, так и находящиеся в процессе становления.

Стоит изучить вопрос практической реализации, который бы основывался на перекрестном применении способов мониторинга кода, а сами способы необходимо сопоставить относительно друг друга. Стоимость использования и внедрения различных средств, их сравнительная эффективность.

Также необходимо уделить внимание проблеме типов в JavaScript, т.е. изучить влияние средств, добавляющих статическую типизацию на конечное качество исходного кода.

Литература

1. PYPL PopularitY of Programming Language [Электронный ресурс]. – Режим доступа: <https://pypl.github.io/PYPL.html> (дата обращения: 20.12.2017).
2. Bryan O'Sullivan. Mercurial: the Definitive Guide. – O'Reilly Media, Inc., 2009. – 284 p.
3. Scott Chacon и др. Pro Git/Chacon. – Apress, 2009. – 288 p.
4. Макконнелл С. Совершенный код. Мастеркласс / Пер. с англ. – М.: Изд-во «Русская редакция», 2010. – С. 493–494.
5. Herczeg Z. et al. Guidelines for JavaScript programs: Are they still necessary // Proceedings of the 11th Symposium on Programming Languages and Software Tools and 7th Nordic Workshop on Model Driven Software Engineering. Tampere University of Technology. – 2009.
6. Стефанов С. JavaScript. Шаблоны. – СПб.: Символ-Плюс, 2011. – 272 с.
7. Бронштейн И.Е. Подход к обнаружению ошибок несоответствия типов в коде на динамических языках программирования // Труды ИСП РАН. – 2013. – С. 67–84.
8. Douglas Crockford – Google+ [Электронный ресурс]. – Режим доступа: <https://plus.google.com/+DouglasCrockfordEsq/posts/MgzNUSTwjRt> (дата обращения: 21.12.2017).



Смирнов Денис Юрьевич

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4208

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: feuway@gmail.com

УДК 004.051

**АНАЛИЗ ИСПОЛЬЗОВАНИЯ СЕРВЕРНОГО «РЕНДЕРИНГА» ПРИ ПОСТРОЕНИИ
СОВРЕМЕННЫХ ВЕБ-ПРИЛОЖЕНИЙ**

Смирнов Д.Ю.

Научный руководитель – д.т.н., профессор Погорелов В.И.

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

В работе произведен сравнительный анализ основных способов «рендеринга» контента веб-приложений, выявлены преимущества и недостатки серверного «рендеринга».

Ключевые слова: серверный «рендеринг», клиентский «рендеринг», JavaScript-фреймворк, Single Page Application.

С незапамятных времен обычным способом для получения HTML-страниц был серверный «рендеринг». Серверный «рендеринг» отлично работал в то время, так как большинство веб-страниц были в основном только для просмотра статических изображений и текста с небольшой интерактивностью.

Сегодня уже многое изменилось. Можно с уверенностью утверждать, что веб-сайты сейчас все больше похожи на «десктопные» приложения [1]. Отправка сообщений, мгновенное обновление информации, интернет-магазины, и многое другое. Интернет теперь – это нечто, намного более продвинутое, чем раньше. Именно поэтому серверный «рендеринг» постепенно уступает место растущему методу «рендеринга» веб-страниц на стороне клиента (Client Side Rendering).

Как работает серверный «рендеринг». Серверный «рендеринг» является наиболее распространенным методом для отображения информации в браузере. Он работает путем преобразования HTML-файлов на сервере в полезную информацию для браузера [2].

Каждый раз при посещении веб-сайта браузер делает запрос на сервер, содержащий содержимое сайта. Запрос обычно занимает всего несколько миллисекунд, но это в конечном счете зависит от множества факторов:

- скорость Интернета;
- местоположение сервера;
- сколько пользователей одновременно пытается получить доступ к сайту.

Как только запрос обрабатывается, браузер получает обратно полностью отображаемый HTML-код и отображает его на экране. Если посетить другую страницу на сайте, то браузер снова сделает еще один запрос за новой информацией [3]. Это будет происходить каждый раз при посещении новой страницы, если браузер не «закэшировал» сайт.

Неважно, будет ли новая страница отличаться всего лишь несколькими элементами от новой, браузер все равно будет формировать запрос для получения новой страницы целиком и заново все «рендерить». Например, если две страницы, по структуре различаются только

парой ссылок, то при переходе между этими страницами браузер будет загружать страницу целиком, а не часть необходимого контента.

Как работает клиентский «рендеринг». Когда рассказывают о «рендеринге» на стороне клиента, говорят о «рендеринге» контента в браузере с помощью языка веб-программирования JavaScript, поэтому вместо того, чтобы получать с сервера все содержимое HTML-документа, мы получаем чистый HTML-документ с файлом JavaScript, который будет «рендерить» контент сайта с помощью браузера [2].

Это относительно новый подход к визуализации веб-сайтов, и он долгое время не был популярным, пока не повысилась производительность JavaScript в браузерах. Данный способ радикально отличается от использования «рендеринга» на стороне сервера, так как сервер теперь только отвечает за загрузку пустого HTML-файла сайта с набором JavaScript-скриптов. Здесь важен сам шаблон. Все остальное обрабатывается на стороне клиента JavaScript-фреймворком.

Ключевая разница заключается в том, что, если нажать на ссылку на странице, чтобы перейти на другую страницу, браузер не будет делать еще один запрос к серверу. «Рендеринг» элементов происходит в самом браузере, и для этого он будет использовать JavaScript, чтобы получить новый контент, и JavaScript-фреймворк, чтобы отобразить только новую контент, а все остальное оставить нетронутым [4]. Это происходит гораздо быстрее, поскольку происходит загрузка только небольшой части страницы, вместо загрузки всей страницы целиком.

Есть несколько нюансов с использованием «рендеринга» на стороне клиента. Поскольку содержимое не отображается до тех пор, пока страница не загрузится в браузере, SEO (Search Engine Optimization) для сайта будет потеряно.

Преимущества и недостатки «рендеринга» на стороне сервера. По сравнению с традиционным SPA (Single Page Application), использующими CSR (Certificate Signing Request), преимуществами серверного «рендеринга» будут:

- лучшее SEO, поскольку поисковые роботы будут видеть полностью «отрендеренную» страницу;
- лучшие показатели времени до отображения контента (time-to-content), особенно при плохом Интернете или на медленных устройствах. Для разметки, «отрендеренной» на сервере, не требуется дожидаться, пока весь JavaScript будет загружен и выполнен, поэтому ваш пользователь увидит полностью «отрендеренную» страницу раньше. Как правило, это приводит к лучшему взаимодействию с пользователем и может быть критичным для приложений, где время до отображения контента напрямую связано с коэффициентом конверсии [5].

Следует учитывать и некоторые недостатки при использовании серверного «рендеринга»:

- ограничения при разработке. Код только для браузера может быть использован лишь в определенных хуках жизненного цикла; некоторые внешние библиотеки могут нуждаться в особой обработке, чтобы иметь возможность запускаться в приложении с серверным «рендерингом» [5];
- более сложные требования по настройке и развертыванию сборки. В отличие от полностью статичного SPA, который может быть развернут на любом статичном файловом сервере, приложение с серверным «рендерингом» требует окружения, где есть возможность запустить сервер Node.js [3];
- повышенная нагрузка на стороне сервера. «Рендеринг» полноценного приложения в Node.js более требователен к ресурсам процессора, чем простая задача статичных файлов, поэтому если ожидается большой трафик, надо быть готовым к соответствующей нагрузке на сервер и использовать стратегии кэширования [3].

Литература

1. The Benefits of Server Side Rendering Over Client Side Rendering [Электронный ресурс]. – Режим доступа: <https://medium.com/walmartlabs/the-benefits-of-server-side-rendering-over-client-side-rendering-5d07ff2cefe8>, своб.
2. Client-side vs. server-side rendering: why it's not all black and white [Электронный ресурс]. – Режим доступа: <https://medium.freecodecamp.com/what-exactly-is-client-side-rendering-and-hows-it-different-from-server-side-rendering-bd5c786b340d>, своб.
3. Server-Side Rendering With React, Node And Express [Электронный ресурс]. – Режим доступа: <https://www.smashingmagazine.com/2016/03/server-side-rendering-react-node-express>, своб.
4. Should I use React Server-Side Rendering? [Электронный ресурс]. – Режим доступа: <https://www.javascriptstuff.com/server-side-render/>, своб.
5. Руководство по серверному рендерингу Vue.js [Электронный ресурс]. – Режим доступа: <https://ssr.vuejs.org/ru/>, своб.

**Сохина Анна Альбертовна**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: 226592@niuitmo.ru

УДК 004

ЭВОЛЮЦИЯ МЕТОДОВ CSS-РАЗМЕТКИ ВЕБ-СТРАНИЦ**Сохина А.А.****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 описаны этапы развития CSS в области разметки веб-страниц, представлены методы создания разметки с помощью CSS-свойств, раскрыты их преимущества и недостатки.

Ключевые слова: верстка, макет веб-страницы, CSS, float, inline-block, flexbox, CSS-grid.

CSS – одна из широкого спектра технологий, одобренных консорциумом W3C под общим названием «Web Standarts». Технология CSS (Cascading Style Sheets) позволила разделить логическую структуру документа от его представления, а также применять для нескольких схожих документов единый стиль оформления.

Первая спецификация CSS1, принятая в 1996 году, была далека от универсального инструмента для верстки. CSS1 на тот момент заменил элементы форматирования из HTML (тэги и
) на язык описания стилей.

Появились свойства блоков, такие как высота и ширина (заменившие одноименные атрибуты в HTML), внутренние (padding) и внешние (margin) отступы и рамки. Также в спецификацию входили ограниченные средства по расположению элементов на странице, такие как float и clear, предназначенные, прежде всего, для обтекания элементов текстом. Однако из-за отсутствия в CSS инструментов, предназначенных для разметки страницы, свойство обтекания стали использовать для создания раскладки страницы.

Разметка на float. Метод раскладки на float являлся первым достойным решением относительно верстки на HTML-таблицах, и, в дальнейшем, даже стал использоваться в большинстве популярных CSS-фреймворков [1]. Но, несмотря на это, данный подход имеет ряд существенных недостатков:

1. при разметке свойством float родительский элемент, в котором расположены float-элементы, «схлопывается» до нулевой высоты, а дочерние элементы выпадают из основного потока. Для решения данной проблемы родительскому блоку требуется задавать так называемый класс .clearfix, или псевдораспорку:

```
.clearfix:after {  
  content: "";  
  display: table;  
  clear: both; }
```

2. если float-элементы, расположенные в ряд, имеют разную высоту, то элементы из следующей строки могут «зацепляться» за элементы первой строки, занимая тем самым свободное место, вместо того чтобы выстраиваться друг за другом в строку. Это

происходит из-за того, что изначально элемент HTML по своей природе всегда стремится к верхнему краю документа, а потом уже влево или вправо, как ему задает свойство float.

Обходить данные проблемы можно путем отмены обтекания у каждого начального элемента в последующей строке. Во фреймворках эта проблема решается наличием элемента-строки (например, класс row в Bootstrap), но тогда нужно четко знать, сколько элементов будет в ряду.

За счет того, что свойство float изначально предназначалось для обтекания картинок текстом, для создания раскладки страницы требуется использование дополнительных приемов и трюков. Также в такой раскладке нет возможности горизонтального и вертикального выравнивания элементов.

Блочно-строчная разметка. В дальнейшем после выпуска спецификации CSS2 в 1998 г., стал доступен еще один способ разметки, использующий блочный, строчный и блочно-строчный контексты форматирования (display: block/inline/inline-block) [2].

Раскладка на блочно-строчных элементах имела ряд преимуществ перед float-раскладкой. Так как эти элементы имеют свойства как блоков, так и строк, их можно выравнивать по горизонтали за счет текстового свойства text-align, и по вертикали за счет блочного свойства vertical-align.

Однако при разметке на inline-block можно столкнуться со следующей проблемой: при выстраивании нескольких блочно-строчных элементов в ряд, между ними образуются пустые текстовые узлы, или whitespaces. Их можно заметить, открыв HTML-код в консоли браузера. Это связано с тем, что inline-block, по сути, имеет свойства обычной буквы (символа), т.е. весь строчный блок составляет всего лишь одну букву в строке, одну единицу, несмотря на то, что может содержать в себе множество других символов и элементов. А следовательно, элементы inline-block, как и простой текст, имеют пробелы, которые и появляются в HTML-разметке.

Существует несколько методов, которые используют разработчики, чтобы удалить эти пробелы из разметки, например, удалить переносы строк в коде HTML, установить в родительском элементе нулевой размер шрифта, добавить элементам отрицательный внешний отступ и др.

Однако данные методы либо засоряют CSS-код лишними свойствами, либо снижают удобочитаемость HTML-разметки. Хотя данный метод раскладки и имеет некоторые преимущества, однако, свойство display: inline-block также, как и float, изначально не предназначено для создания сложных разметок документа, поэтому имеет весьма скудные возможности по настройке.

CSS3 и Box Alignment Module. Ситуация в области разметки веб-страницы кардинально изменилась с выходом новой спецификации CSS3. Данная спецификация является самой масштабной по сравнению с CSS1, CSS2 и CSS2.1. В отличие от предыдущих версий спецификация разбита на модули, разработка и развитие которых идет независимо.

Значительную роль в области разметки страницы сыграл модуль CSS Box Alignment Module Level 3 [3], который ввел новые свойства выравнивания элементов по главной и поперечной осям, выравнивания элементов внутри контейнера, выравнивания отдельно взятых элементов. Также данная спецификация ввела и новые значения свойств, что позволило настраивать раскладку элементов еще проще. Например, значения flex-start/flex-end выравнивают элементы от начала или от конца главной оси соответственно. А благодаря значениям space-around/space-between элементы выравниваются по главной оси, распределяя свободное место вокруг себя или между собой. На основе данного модуля выравнивания базируются два вида раскладки, разработанных в стандарте CSS3 – Flexible Box Layout Module и Grid Layout Module.

1. Flexbox. Одним из основных понятий во flex-разметке являются главная и поперечная оси. Направление осей можно изменять с помощью свойства flex-direction: column/row, заданного для контейнера с элементами [4]. С помощью свойства flex-wrap можно управлять выводом элементов – с переносом блоков в следующую строку или без него.

Создавать гибкую раскладку на flexbox позволяют следующие свойства:

- flex-grow: определяет, насколько будет растягиваться элемент относительно других элементов в контейнере при увеличении размеров экрана;
- flex-shrink: определяет, насколько будет сужаться элемент относительно других элементов в контейнере при уменьшении размеров экрана;
- flex-basis: указывает исходный размер элемента, прежде чем свободное пространство распределится в соответствии с факторами гибкости.

Одним из преимуществ flex-разметки по сравнению с предыдущими: высота всех элементов автоматически подстраивается под наибольший элемент в строке. Настроить элементы таким образом при float- и inline-block-раскладках невозможно.

Также в раскладке на flexbox можно изменять порядок блоков за счет свойства order. Данное свойство может быть весьма полезным при адаптации веб-страницы под разные разрешения экранов, так как зачастую требуется изменить порядок отображения элементов на мобильных устройствах.

2. Grid Layout. Появление флексбоксов несомненно упростило решение большого количества задач верстки, но данная технология пригодна для построения одномерных макетов, т.е. по одной оси. По этой причине вскоре был введен новый модуль разметки страницы (Grid Layout Module) [5], который позволяет строить макет веб-страницы одновременно по двум осям.

Для того чтобы создать сетку в документе достаточно задать родительскому контейнеру display: grid, настроить размеры и количество колонок и строк с помощью свойств grid-template-columns и grid-template-rows. Чтобы разместить элементы внутри сетки, им задаются свойства grid-column и grid-row.

В CSS-Grid вводится новая относительная единица измерения – fr, определяющаяся как фракция или часть, т.е. если у одной колонки задана ширина в 1fr, а у второй – 2fr, следовательно, вторая колонка будет в 2 раза большей ширины, чем первая. Данная единица помогает легко создавать гибкие сетки.

Также, помимо большого количества настроек, предоставляемых модулем Box Alignment, в CSS-Grid также введены новые функции:

- repeat(): определяет повторяющиеся колонки сетки, принимает два аргумента: количество повторений и размер колонки;
- minmax(): задает минимум, до которого можно ужимать полосу, и максимум, на который она может растянуться, если хватит места;
- fit-content(): принимает параметр, который указывает максимальный размер полосы сетки.

В CSS-Grid также существует такое понятие как «grid-area». Это пространство внутри контейнера с сеткой, в которое может быть помещен один или больше элементов сетки. Этот элемент может состоять из одной или более ячеек сетки. Для работы с grid-areas существуют два свойства:

1. grid-template-areas: свойство для контейнера сетки. Определяет шаблон сетки, ссылаясь на имена областей, которые заданы с помощью свойства grid-area. Повторение названия области приводит к тому, что содержимое охватывает эти ячейки. Точка означает пустую ячейку. Сам синтаксис предоставляет визуализацию структуры сетки;
2. grid-area: свойство для дочерних элементов сетки. Дает название элементу сетки, чтобы можно было ссылаться на него с помощью шаблона, созданного через grid-template-areas.

Таким образом можно разметить страницу, используя шаблон grid-area:

```
grid-template-areas: "header header"
```

```
"menu content"  
"footer footer";
```

Одним из основных преимуществ Grid Layout является то, что подобная раскладка полностью не зависит от кода HTML и порядка следования элементов в нем, так как положение каждого элемента на странице задается в CSS.

Выводы. Несомненно, за последние годы стандарт CSS3 произвел настоящую революцию в области разметки веб-страниц: возникновение таких технологий как Flexbox и CSS-Grid упростило решение огромного количества задач в верстке и позволило избежать проблем, существовавших ранее в разметках на float и inline-block. На данный момент Flexbox поддерживается практически наравне с inline-block (более 95%), что позволяет применять данную технологию для большинства задач при верстке. Что касается CSS-Grid, так как данный модуль является достаточно новым (февраль-март 2017 года), ее поддержка браузерами пока не достаточно высока (76,7%). Также стоит отметить, что излишне частое использование Flexbox и Grid на странице может давать нагрузку на производительность, так как эти модули содержат большое количество настроек, поэтому не стоит использовать Flexbox и Grid в тех задачах, которые можно решить с помощью float или inline-block, чтобы не перегружать браузер.

Литература

1. CSS layout – float and clear [Электронный ресурс]. – Режим доступа: https://www.w3schools.com/css/css_float.asp, своб.
2. CSS layout – inline-block [Электронный ресурс]. – Режим доступа: https://www.w3schools.com/css/css_inline-block.asp, своб.
3. CSS Box Alignment Module Level 3 [Электронный ресурс]. – Режим доступа: <https://www.w3.org/TR/css-align-3>, своб.
4. CSS Flexible Box Layout Module Level 1 [Электронный ресурс]. – Режим доступа: <https://www.w3.org/TR/css-flexbox-1>, своб.
5. CSS Grid Layout Module Level 1 [Электронный ресурс]. – Режим доступа: <https://www.w3.org/TR/css-grid-1>, своб.

**Ставрова Владислава Сергеевна**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4205Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: vlstavrova@mail.ru

УДК 004.921

**ТЕХНОЛОГИИ ОПТИМИЗАЦИИ ТЕКСТУР ДЛЯ ИГРОВОЙ ТРЕХМЕРНОЙ
ГРАФИКИ****Ставрова В.С.****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 рассмотрены два основных метода оптимизации текстур: использование атласов и тайлов, даны их определения. Выявлены способы создания текстурных атласов, ручные и автоматические, пояснено, каким образом происходит оптимизация конечного продукта, проанализировано влияние на визуальное качество игры. Обозначены особенности работы с тайловой графикой, выявлены способы ее использования без потерь в качестве.

Ключевые слова: оптимизация, текстуры, трехмерная графика, тайлы, атласы, игры.

При работе с игровой трехмерной графикой процесс рендеринга играет существенную роль. На фоне всех процессов игры, происходящих в реальном времени, он является наиболее сложным и трудоемким. Это обуславливает важность максимального использования любых методов для уменьшения временных затрат на него. При сокращении времени происходит увеличение частоты кадров (FPS), либо, при нормальной исходной производительности, появляется возможность включить дополнительные игровые элементы.

Понятия Time per frame и Frame per second (FPS) – это время, которое затрачивается на работу с одним кадром и его рендеринг, и обратная величина, означающая количество кадров в секунду [1].

Текстурные атласы являются эффективным средством для снижения временных затрат на процесс рендеринга. Это набор текстур для разных объектов, объединенных в едином файле изображения. Для их создания используются ручные (при работе с исходными текстурами) и автоматические методы, осуществить которые помогают программы сборки.

Ручное создание осуществляется следующим образом:

1. UV-развертки трехмерных моделей «уменьшаются» в трехмерном редакторе и помещаются на единую развертку при максимальном использовании пустого места;
2. в графическом редакторе создается изображение требуемого размера (как правило, кратного степени двойки: 512×512, 1024×1024, 2048×2048). Затем текстуры помещаются на него в соответствии с новой UV-разверткой.

Зачастую можно сразу помещать текстуры в атлас при работе с трехмерной моделью.

Примерами инструментов для создания атласов являются такие программы, как Sprite Packer и Texture packer. Sprite Packer встроен в игровой движок Unity, а Texture Packer – это отдельная платная программа.

Текстурные атласы оптимизируют игру следующим образом: для элементов интерфейса и сцены, имеющих индивидуальную текстуру, вызывается отдельный метод отрисовки для каждого из фрагментов. Каждый метод при вызове требует дополнительное

время на обработку, поэтому скорость рендеринга сокращается. Получается цепочка факторов, следующих один из другого:

1. увеличение количества методов отрисовки;
2. увеличение времени для рендеринга;
3. уменьшение FPS.

В итоге игра получается либо с низким показателем FPS, либо с меньшим количеством объектов, что влияет на качество игрового процесса, либо на эстетическую картину.

Можно сделать вывод, что если объединить все текстуры в единый файл, то они будут отрендерены в один проход, что значительно увеличит производительность [2].

Использование текстурных атласов можно назвать способом оптимизации, позволяющим сохранить или даже улучшить визуальное качество изображения. При помещении текстур в атлас имеется возможность сохранить их исходные размеры в зависимости от требований, однако производительность увеличится, а возможность добавить больше элементов в игру улучшит ее эстетические качества.

Иной эффективный метод оптимизации – это использование тайловых текстур. Тайловая графика – это способ, позволяющий отобразить крупные элементы поверхности. Как правило, это уровни в двумерных компьютерных играх, либо текстуры крупных трехмерных объектов.

Название тайловая графика связано с тем, что изображение создается по аналогии с картиной из изразцов, из небольших фрагментов (паттернов) одного размера. Такая технология дает возможность создать визуализацию огромных пространств, как двумерных, так и трехмерных (повторяющиеся текстуры на трехмерных объектах). Памяти при этом расходуется в минимальном количестве. Тайловая текстура создается бесшовной, чтобы скрыть стыки между фрагментами.

При работе с тайловыми текстурами не только оптимизируется конечный продукт, но и экономится время и ресурсы при разработке. Для таких объектов, как пол, стена и иных, крупных по площади, создается текстура без швов на всю поверхность. Создание единой уникальной текстуры для, например, монотонной черепичной крыши, будет излишне трудоемким процессом. Важно учитывать, что пользователь в дальнейшем не должен заметить разницу между краями текстуры [3].

Часто для создания тайловой текстуры используются фотографии объекта из реального мира, такого как кирпичная стена. Все края изображения должны совпадать с каждым прилегающим к ним. В итоге получается единая большая текстура необходимых габаритов, без визуальных стыков.

Простого использования фотографии в качестве текстуры, повторяющейся на объекте, недостаточно. Например, на фотографии плиточной стены, левая сторона не будет совпадать с правой, а нижняя не будет совпадать с верхней. Для того чтобы текстура стала правильной, необходима ее обработка в графическом редакторе, например, в Photoshop. Также существуют и иные способы создания тайловых текстур, вплоть до их рисования.

Использование тайлов безусловно решает задачу оптимизации, однако, качество изображения ухудшается из-за возникающей монотонности: глаз пользователя легко замечает повторяющиеся элементы. Но существуют способы обойти этот дефект:

1. вместо единого изображения тайла на весь объект создается его несколько вариаций. Эти варианты беспорядочно помещают таким образом, чтобы скрыть однообразие множества фрагментов;
2. в дополнение к текстуре создаются отвлекающие элементы. Технически, это могут быть спрайты, либо дополнительные тайлы. Существует возможность наложить дополнительный тайл как верхний слой над основным;
3. использование клеточной механики игры как оправдание для повторяющихся фрагментов;
4. оправдание монотонности графическим стилем, либо идентичностью объектов. Удачным примером такой игры является Minecraft [4, 5].

Таким образом, при грамотном использовании тайловых текстур, такой способ оптимизации обеспечит сохранность эстетических качеств.

Литература

1. Окно Rendering Statistics [Электронный ресурс]. – Режим доступа: <https://docs.unity3d.com/ru/530/Manual/RenderingStatistics.html/> (дата обращения: 13.01.2018).
2. Использование атласа текстур для оптимизации вашей игры [Электронный ресурс]. – Режим доступа: <http://faircat.ru/ispolzovanie-atlasa-tekstur-dlya-optimizatsii-vashej-igry/> (дата обращения: 13.01.2018).
3. Тайлинг ландшафтов [Электронный ресурс]. – Режим доступа: <http://www.gamedev.ru/code/articles/?id=4208> (дата обращения: 14.01.2018).
4. Аксенов А.В., Гребенникова Е.Л. О технологиях создания текстурированных трехмерных моделей для игровых приложений // Научная сессия ГУАП: сб. статей. – 2015. – С. 148–154.
5. Сеидова А.С., Сухоплюева В.С. Разработка мобильных игр с помощью Unity 3D // Информационно-телекоммуникационные системы и технологии: сб. статей. – 2014. – С. 294–295.



Тодоровская Яна Васильевна

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: correct.V@yandex.ru



Шуклин Дмитрий Анатольевич

Год рождения: 1979

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: do@limtu.ru

УДК 004.5

**ОБЗОР ПОДХОДОВ К ПРОЕКТИРОВАНИЮ ПОЛЬЗОВАТЕЛЬСКИХ
ИНТЕРФЕЙСОВ**

Тодоровская Я.В., Шуклин Д.А.

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

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

В области проектирования пользовательских интерфейсов существует ряд проблем, таких как эргономичность, удобство использования, простота навигации по контенту, показатели конверсии. Существует ряд подходов, которые будут решать перечисленные проблемы. В работе был проведен обзор этих подходов, некоторые подходы ориентированы на потребности пользователей и их требования, некоторые направлены на проектирование продукта на основе данных: исследований, тестов, проверки гипотез, Big Data.

Ключевые слова: пользовательский интерфейс, проблемы разработки интерфейса, подходы к разработке интерфейса.

Разработка пользовательского интерфейса является неотъемлемой составляющей большинства проектов, так как пользователь взаимодействует с системой именно через интерфейс. Эффективность разрабатываемого продукта будет зависеть от того, насколько эффективна и полезна работа пользователя с системой.

Интерфейс – совокупность возможностей, способов и методов взаимодействия двух систем.

Интерфейс пользователя, он же пользовательский интерфейс – это та часть интерфейса интернет-ресурса/портала, с которой пользователи взаимодействуют, то что они напрямую видят, щупают и т.п.

По оценкам специалистов проектирование, разработка, модификация, сопровождение пользовательского интерфейса занимает достаточно большое количество человеко-часов. В настоящее время в области разработки существует ряд проблем:

1. усложнение интерфейсов, связанное с тем, что условия предметной области постоянно меняются в связи с изменением и развитием бизнес-процессов, а также с увеличением функциональности проекта;
2. интерфейс не соответствует требованиям и ожиданиям пользователей;

3. исключение постоянного взаимодействия между заказчиком и подрядчиком;
4. интерфейс не ориентирован на деятельность пользователя.

Для решения вышеперечисленных проблем и эффективности результата разработки используют различные подходы к проектированию пользовательских интерфейсов, которые в какой-то степени помогают решать вышеперечисленные проблемы [1].

Дизайн, ориентированный на пользователя (User-centered design (UCD)), возник в начале 1980-х годов – это процесс проектирования, который ориентирован на потребности и требования пользователя. Этот метод был определен разными авторами и в ряде положений стандартов ISO, таких как № 13407 (проектирование интерактивных систем, ориентированных на пользователей) и TR 18529 (этапы внедрения проектов, ориентированных на пользователей). Основные положения UCD изложены в стандарте ISO (ISO 9241-210, 2010). Постоянное применение факторов, оказывающих влияние на эргономику, технику юзабилити, эстетику, производительность – это то, что поддерживает UCD. Этому подходу характерно постоянное взаимодействие с пользователем, для того чтобы получить представление об их целях и задачах, по возможности, для эффективности взаимодействия и разработки продукта необходимо вовлечение в деятельность пользователя, что и является решением четвертой проблемы.

На протяжении всего жизненного цикла проекта для успешной работы пользователя, дизайнеру необходимо соблюдать баланс между вышеперечисленными факторами. Это достигается тщательной проработкой интерактивного взаимодействия на каждом этапе разработки USD-подхода:

1. исследование. На данном этапе дизайнеры пытаются понять контекст, в котором может использоваться система. Впоследствии будут указаны требования пользователей;
2. проектирование. На данном этапе определяются цели разработки, разрабатываются сценарии, впоследствии проводится оценка сценариев с ожиданиями пользователей. После чего производится концептуальное проектирование, определение информационных потоков, выбор платформы, на которой будет разрабатываться проект;
3. реализация. В результате разрабатывается прототип на бумажных и (или) интерактивных макетах, создание проекта с учетом возможности изменения дизайна;
4. тестирование и оценка. Проводится тестирование продукта различными пользователями с учетом их ожиданий и требований.

Так как создание пользовательского интерфейса процесс итеративный, то такие этапы как реализация и тестирование маловероятно, что будут завершены за один подход, поэтому проводятся итерации этапов, пока результаты оценки не будут удовлетворительными.

Другими словами, UCD направлен на удовлетворение потребностей пользователей, делая это посредством сочетания исследований (например, опросов и интервью) и генеративных (например, «мозгового штурма») методов и инструментов.

Команда дизайнеров должна включать специалистов по нескольким дисциплинам (например, этнографов, психологов, инженеров по программным и аппаратным средствам), а также экспертов домена, заинтересованных сторон и самих пользователей. В этом подходе важно, чтобы было привлечение пользователей (повышение конверсии), а также долгосрочный мониторинг использования проекта.

Одним из наиболее ярких представителей проектирования, ориентированного на пользователя – Дональд Норман. Однако в настоящее время, он является критиком данного подхода, суть его критики состоит в том, что проектирование в рамках подхода, ориентированного на пользователя, слишком сконцентрировано на удовлетворении просьб и желаний пользователей. Разработчики и аналитики составляют подробнейшее описание требований и желаний пользователей, но не уделяют должного внимания перипетиям той деятельности, в которой занят пользователь. Дональд Норман утверждает, что если

некоторый артефакт идеально приспособлен к выполнению некоторой деятельности, то пользователь будет готов потратить некоторое время, чтобы приспособиться к нему [2].

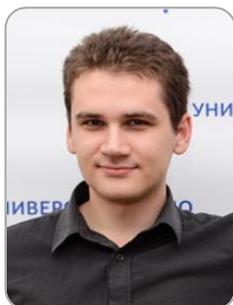
- системный подход (System) – подход, в котором пользователь находится во взаимодействии, является маленькой частью системы «человек-компьютер»;
- деятельностный подход (Activity-centered design) – подход, в основе которого лежит теория деятельности, представляющая компьютер в качестве инструмента, с помощью которого пользователь решает различные задачи. В этом подходе именно деятельность человека влияет на интерфейс. После целостного изучения деятельности, происходит оптимизация отдельных моментов;
- итеративный подход (Agile) – метод последовательного приближения. В данном методе происходит тесное взаимодействие «Заказчик – подрядчик» [3].

Дизайн на основе данных (Data-Driven Design (DDD)) – это проектирование дизайна продукта на основе данных: исследований, тестов, проверки гипотез, Big Data, тем самым определяя наиболее эффективную схему проектирования. DDD закрывает недостатки традиционного дизайн-подхода как на стороне подрядчика, так и на стороне заказчика. Дизайн, разработанный посредством данных, будет точно нацелен на клиентов, здесь заказчику не надо будет заказать сайт под ключ и полагается на опыт дизайнера, его понимании задач, навыках, изобретательности, чувстве вкуса. DDD – это уже не поиск решения главной проблемы и тестирование дизайн-идей. Этот метод охватывает всех пользователей, анализирует данные об их интересах, стиле жизни, ключевых проблемах и социальном статусе [4].

В заключении следует сказать, что подходы к проектированию развиваются в соответствии со временем роста информационных технологий, они имеют тенденцию устаревать или же преобразовываться в другие методы. Перечисленные подходы нельзя рассматривать изолированно. Их сочетание и уместное применение позволяет разрабатывать интерфейсы, с которыми пользователи смогут успешно взаимодействовать. С другой стороны каждый из них сосредоточен на одном аспекте интерфейса и необходим более широкий, единый, комплексный подход [5].

Литература

1. Грибова В.В., Клещев А.С. Концепция разработки пользовательского интерфейса на основе онтологий. Ч. 1. Инструментарий для разработки пользовательского интерфейса. Основная идея подхода. – Владивосток: ИАПУ ДВО РАН, 2003. – 24 с.
2. Норман Д.А. Дизайн привычных вещей. – Изд-во: Вильямс, 2006. – 384 с.
3. Клещев А.С. Методы и средства разработки пользовательского интерфейса. Современное состояние [Электронный ресурс]. – Режим доступа: <http://swsys.ru/index.php?page=article&id=765>, своб.
4. Lewis C., Rieman J. Task-Centered User Interface Design A Practical Introduction [Электронный ресурс]. – Режим доступа : <http://www.hcibib.org/tcuid/chap-0.html>, своб.
5. Купер А. Об интерфейсе. – Изд-во: Символ-Плюс, 2009. – 688 с.

**Туркунов Александр Алексеевич**

Год рождения: 1991

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4206Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: snowfayu@gmail.com

УДК 004.92

**ОЦЕНКА ПРАКТИЧЕСКОГО ВНЕДРЕНИЯ ТЕХНОЛОГИЙ VR И AR
ДЛЯ ДИСПЕТЧЕРИЗАЦИИ И ОТСЛЕЖИВАНИЯ СИСТЕМ/ОБОРУДОВАНИЯ
В ЗДАНИЯХ ОБЩЕГО ПОЛЬЗОВАНИЯ**

Туркунов А.А.

Научный руководитель – к.ф.-м.н. Соловьев И.А.

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

В работе проведена оценка практического внедрения технологий VR и AR для диспетчеризации и отслеживания систем в зданиях общего пользования на примере компании UNIOS, рассмотрены их основные достоинства и недостатки.

Ключевые слова: технологии, виртуальная реальность, дополненная реальность, оборудование, диспетчеризация, контроль, перспективы.

На сегодняшний день ни одно здание или инфраструктура не обходятся без высокотехнологичных инженерных систем на своей территории. Пожарные системы, электронные замки, воротные системы, системы вентиляции и отопления, электросети – это всего лишь малая часть эксплуатируемого оборудования, которое в свою очередь нуждается в постоянном и тщательном обслуживании. Это еще более актуально для узкоспециализированного оборудования заводов, фабрик и прочих промышленных объектов. Внедрение новых информационных технологий не обходит стороной и такую отрасль. Не стало исключением появление технологии виртуальной и дополненной реальности (VR, AR) в архитектуре, планировании и строительстве, что, в свою очередь, открыло нам новые возможности для улучшения и облегчения работы с инженерными системами. Применение технологий дополненной реальности помогает визуализировать различные процессы (в том числе технологически сложные), показать предполагаемый конечный результат, помочь разобраться в работе устройства [1] (рис. 1).

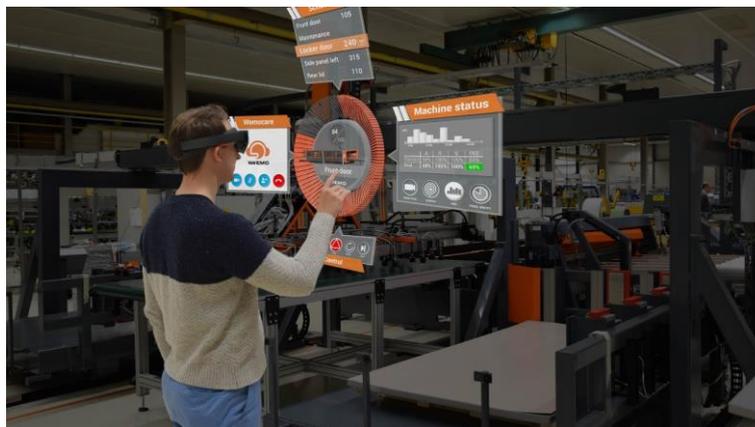


Рис. 1. Использование голографического интерфейса-справки

Преимущества. При работе с высокотехнологичным оборудованием, интуитивно понятная визуальная составляющая может стать мощнейшим инструментом, которое исключит большинство проблем, связанных с потерей времени, невнимательностью и возможной некомпетентностью персонала. Каждое устройство, имеющее виртуальную проекцию, совпадающую с реальным прототипом в пространстве, и оборудованное интерфейсом-библиотекой, будет передавать вам информацию о своем состоянии! Не придется долго выяснять причины поломок или сбоев, кроме того, будут приведены конкретные рекомендации и инструкции по ремонту и замене элементов, вышедших из строя.

Сотрудник обслуживающего персонала сможет легко ориентироваться на местности и находить нужные инженерные системы, и при случае может быть информирован полезной информацией, которая поможет ему в его работе с оборудованием [1] (рис. 2).

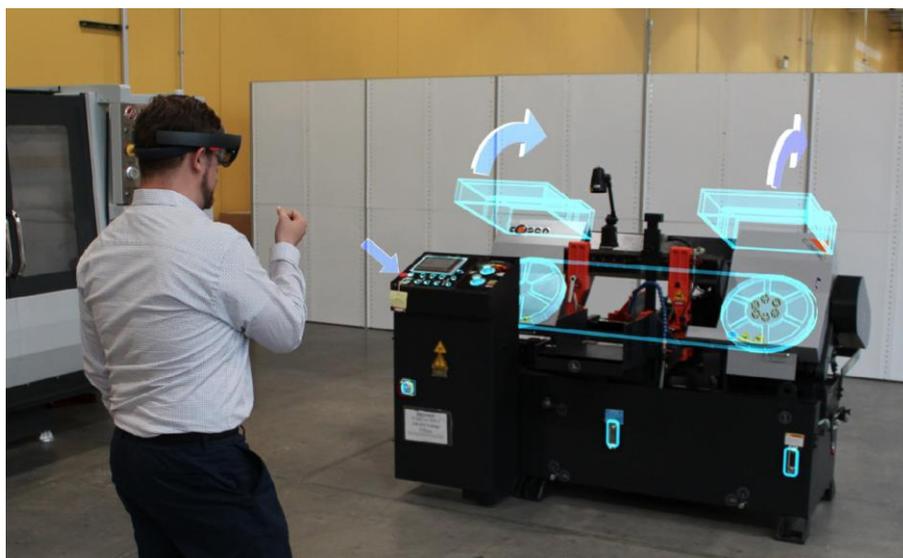


Рис. 2. Использование голографического интерфейса-справки при работе на станке

Главная проблема. Данная реализация подобной системы крайне затруднительна. Каждое устройство/система/оборудование нуждается в реализованной виртуальной модели с понятным интерфейсом, который бы отвечал требованиям работы с данным устройством/системой/оборудованием. Необходимо провести большую работу по написанию инструкций и программ. Помимо этого, пока не существует подходящего устройства для работы в виртуальном пространстве, которое может работать достаточно долго и при этом было бы достаточно мобильно, для работы на любом объекте. На сегодняшний день дальше всего в данной области продвинулись Microsoft со своим продуктом «HoloLens».

Microsoft HoloLens – является первым успешным проектом компании Microsoft по созданию устройства смешанной реальности и первый продукт, работающий в подсистеме Windows Holographic и операционной системе Windows 10.

На данный момент это устройство продается по цене в 3000 долларов исключительно для разработчиков США и Канады и еще в 29 европейских регионах. А в России их пока нет. Сейчас ведется работа над обновлением устройства, его программного обеспечения и функционала, релиз запланирован на начало 2018 года.

Его главное отличие от существующих аналогов заключается в автономности. Устройство работает на Windows и его батареи хватает на 5–6 ч непрерывной работы. Соответственно, оно работает без подключения к компьютеру посредством кабеля, и представляет собой автономное устройство, на котором и производятся все вычисления с помощью CPU и GPU.

Голограммы и смешанная реальность в концепции компании Microsoft позволяют дополнить окружающее пространство трехмерными объектами, которые будут не просто «болтаться в воздухе», как в обычном варианте дополненной реальности, а органично вписываться в окружение и взаимодействовать с ним. HoloLens имеет очень высокую точность позиционирования и учитывает особенности человеческого восприятия комбинации из реальных объектов и изображения на дисплеях. Именно благодаря этому голограммы выглядят такими стабильными [2].

Вывод. Несмотря на все сложности, дальнейшее развитие данного направления, как и всех отраслей систем дополненной и виртуальной реальностей, неизбежен. Прогресс не стоит на месте, технологии VR и AR совершенствуются, и все больше отраслей промышленности начинают использовать открывающиеся возможности. Как утверждают эксперты, уже в скором будущем все участники строительства будут получать необходимую информацию о стройке прямо на поверхности своих очков или линз. Кипы чертежей заменят технологии дополненной реальности, которые уже и на сегодняшний день способны помочь инженерам в работе [3–7].

Литература

1. Scope AR's WorkLink Platform Brings Smart Instructions To HoloLens [Электронный ресурс]. – Режим доступа: <https://www.vrfocus.com/2017/05/scope-ars-worklink-platform-brings-smart-instructions-to-hololens/> (дата обращения: 21.12.2017).
2. Microsoft HoloLens [Электронный ресурс]. – Режим доступа: <https://www.ferra.ru/ru/periphery/news/2017/11/01/microsoft-hololens-expands-29/> (дата обращения: 21.12.2017).
3. Wellner P., Mackay W. & Gold R. Eds. Special issue on computer augmented environments: back to the real world // Communications of the ACM. – 2012. – V. 36. – № 7. – P. 24–27.
4. Россохин А., Измагурова В.. Виртуальное счастье или виртуальная зависимость // Личность в измененных состояниях сознания. – 2004. – С. 516–523.
5. Войскунский А.Е., Смылова О.В. Психологическое применение систем виртуальной реальности // Интернет и современное общество. Труды IX Всероссийской объединенной конференции. – 2006 [Электронный ресурс]. – Режим доступа: <http://www.conf.infosoc.ru/2006/thes/Voisk&Smyslova.pdf>, своб.
6. Клайн Э. Первому игроку приготовиться. – Астрель, Neoclassic, 2013. – 480 с.
7. 9 сфер применения виртуальной реальности: размеры рынка и перспективы [Электронный ресурс]. – Режим доступа: <https://vc.ru/p/vr-use> (дата обращения: 21.12.2017).



Фирсова Маргарита Алексеевна

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: margaret2009@rambler.ru



Государев Илья Борисович

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: goss@corp.ifmo.ru



Сокуренок Юрий Андреевич

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», к.т.н., доцент

e-mail: kpd@limtu.ru

УДК 004

**ОБЗОР ПАТТЕРНОВ МЕНЕДЖМЕНТА ЗАВИСИМОСТЕЙ В JAVASCRIPT-
ФРЕЙМВОРКАХ**

Фирсова М.А., Сокуренок Ю.А., Государев И.Б.

Научный руководитель – к.т.н., доцент Сокуренок Ю.А.

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

В работе исследован вопрос применения паттернов менеджмента зависимостей для разработки веб-приложений, основанных на JavaScript-фреймворках. Рассмотрены основные примеры паттернов менеджмента зависимостей, которые используются при веб-разработке и могут быть применены для разработки приложений с помощью языка программирования JavaScript.

Ключевые слова: паттерны, Dependency Injection, Inversion of Control, Service Locator, паттерны управления зависимостей, инверсия управления, JavaScript.

В сфере веб-технологий стремительно набирает обороты популярность веб-приложений, что актуализирует проблему разработки более новых приложений на основе готового шаблона кода, имеющего независимые модули, которые могут быть свободно перенесены в код другого приложения, не вызывая конфликтов между модулями в новом приложении.

В настоящее время растет спрос на создание более сложных веб-приложений, такая тенденция заставляет разработчиков задумываться о создании готовых шаблонов кода. Такие шаблоны, называемые паттернами проектирования, можно повторно применять для

разработки в будущем других приложений, также такие шаблоны должны изолировать модули разрабатываемых приложений друг от друга, создавая слабосвязанный код [1].

При разработке архитектурно нагруженных веб-приложений, которые имеют строгие зависимости между функциональными модулями, могут возникнуть проблемы, связанные с управлением модулями и повторным использованием модулей для других веб-разработок.

Для решения описанной проблемы существуют паттерны менеджмента зависимостей, которые позволяют создавать слабосвязанный код при реализации многомодульных нагруженных приложений на разных объектно-ориентированных языках программирования, таких как Java, C#, .NET, JavaScript [2, 3].

Паттерны менеджмента зависимостей в современном мире веб-разработки известны, как подход Inversion of Control, который предполагает прагматичный способ обеспечения свободного соединения зависимых модулей, которые в совокупности составляют программное приложение, тем самым подразумевая использования паттернов менеджмента зависимостей.

Понятие Inversion of Control не является новым в сообществе JavaScript-разработчиков, но, при его популярности в других языках объектно-ориентированного программирования (Java, C++, .NET), использование данного паттерна при разработке JavaScript-приложений встречается довольно редко.

Несмотря на все свои очевидные плюсы, инверсия управления является лишь общим набором рекомендаций, что предполагает наличие более конкретных паттернов. Таких паттернов два: Service Locator и Dependency Injection.

В работе [4] нашего соотечественника С.В. Теплякова говорится, что с появлением Service Locator началось активное развитие практик инверсии управления во всем мире. Суть паттерна Service Locator заключается в том, что, вместо создания конкретных сервисов напрямую, с помощью ключевой команды «new», используется специальный «фабричный объект», который отвечает за создание и нахождение всех необходимых для работы сервисов.

В общем случае при использовании паттерна Service Locator на рисунке, а, ClassA использует объект локатор, который занимается получением сервисов. Сервис локатор может быть представлен в виде статического класса с набором статических методов, также Service Locator может быть определен в виде интерфейса для упрощения тестирования. Назначение паттерна Service Locator состоит в том, чтобы разорвать жесткую связь между классом и его сторонними сервисами. Это достигается за счет добавления специального класса Locator, который будет нести ответственность за связь со вспомогательными классами.

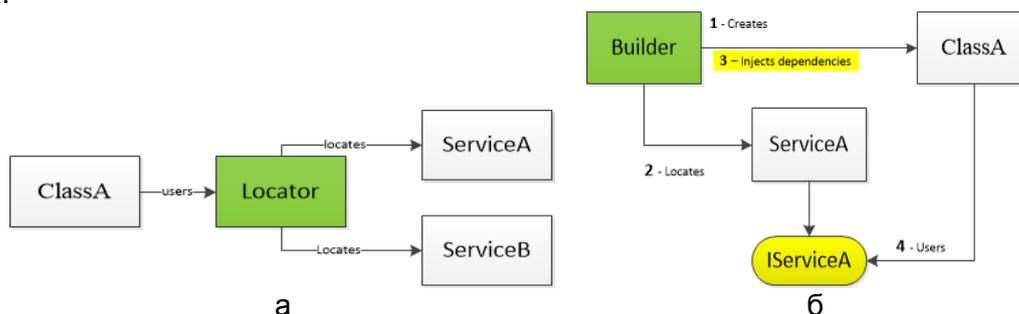


Рисунок. Паттерн Service Locator (а); схема паттерна Dependency Injection (б)

В подавляющем большинстве случаев исследования на тему паттернов Inversion of Control наших соотечественников ссылаются на исследования зарубежного ученого Мартина Фаулера. Самой популярной статьей М. Фаулера является статья «Inversion of Control Containers and the Dependency Injection pattern», которая была написана ученым более десяти лет назад и относилась к паттернам инверсии управления в языке Java.

Именно М. Фаулером было введено более конкретное, в отличие от инверсии управления, по его мнению, понятие Dependency Injection, что переводится как внедрение зависимостей [3].

Внедрение зависимостей (DI, Dependency Injection) – это механизм передачи классу его зависимостей. Паттерн DI позволяет избежать жестко закодированных зависимостей между классами при разработке модульного приложения. А именно, паттерн внедрения зависимостей делает модули веб-приложения свободно связанными. Такой подход позволяет разбивать код на независимые модули, которые можно будет использовать в будущем при работе над другими веб-проектами.

Паттерн DI позволяет внедрять зависимости в объекты непосредственно во время выполнения программного кода, исключая необходимость жесткого кодирования этих зависимостей. В общем случае на рисунке, б, модуль builder создает экземпляр класса. После создания класса classA, модуль builder внедряет параметры для создания зависимостей в этот класс, тем самым предоставляя различные API (Application Programming Interface), через которые classA, сможет получать от сервисов необходимые ему услуги. В данном случае IServiceA является интерфейсом сервиса ServiceA для класса classA, а именно, builder отправляет сообщение классу classA о том, какие зависимости данный класс может использовать. Также модуль Builder сообщает сервису, какие методы тот может передать для класса classA через интерфейс IServiceA, когда класс classA к нему обратится.

Исходя из представленной на рисунке, б, реализации паттерна внедрения зависимостей, можно сказать, что builder является независимым модулем, соответственно, данный модуль можно использовать как плагин и переносить между проектами.

Марк Симан выделяет три вида паттернов внедрения зависимостей [5, С. 121]:

- внедрение через конструктор (Constructor Injection);
- внедрение через метод (Method Injection);
- внедрение через свойство (Property Injection).

Через конструктор (Constructor Injection) передаются все зависимости, требуемые некоторому классу, в качестве параметров конструктора, представленных в виде интерфейсов или абстрактных классов. Передача зависимостей через конструктор является предпочтительным способом внедрения зависимостей, поскольку в этом случае проще всего обеспечить четкий контракт между классом и его клиентами.

Property Injection используется для установления необязательных зависимостей, для которых существует значение по умолчанию. Суть такого внедрения зависимостей состоит в том, что необязательная зависимость, требуемая некоему классу, может быть переопределена вызывающим кодом путем установки ее через свойство.

При использовании Method Injection передаются зависимости, необходимые только для одного метода класса, при этом зависимости не нужно передавать остальным.

Исходя из вышеописанных примеров паттернов менеджмента зависимостей, можно утверждать, что подход инверсии управления (IoC) является широко применяемым методом для проектирования различного программного обеспечения, для повторного использования и обслуживания.

Подход Inversion of Control предлагает прагматичный способ обеспечения свободного соединения зависимых модулей, которые в совокупности составляют программное приложение. Благодаря преимуществам, полученным в результате повторного использования, IoC в настоящее время является широко применяемым принципом во многих объектно-ориентированных и компонентных модулях. Следуя этому принципу, свободное сцепление модулей обеспечивается инвертированием обязанностей, которые приводят к зависимостям между компонентами приложения.

В современных исследованиях чаще вместо Inversion of Control встречается паттерн Dependency Injection [3], который обеспечивает превосходную альтернативу оптимизации при разработке современных веб-программ.

Литература

1. Гамма Э., Хелм Р., Джонсон Р., Влссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб.: Питер, 2015. – 368 с.
2. Бучельников И.В. Так все-таки, что такое Inversion of control // RSDN Magazine. – 2012. – № 3. – С. 54–62.
3. Fowler М. Inversion of Control Containers and the Dependency Injection pattern [Электронный ресурс]. – Режим доступа: <https://martinfowler.com/articles/injection.html/>, своб.
4. Тепляков С.В. Паттерны внедрения зависимостей // RSDN Magazine. – 2012. – № 4. – С. 14–22.
5. Симан М. Внедрение зависимостей в .NET. – СПб.: Питер, 2014. – 461 с.



Фомин Антон Артурович

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4108

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: fominanton.070695@gmail.com



Сокуренько Юрий Андреевич

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», к.т.н., доцент

e-mail: kpd@limtu.ru

УДК 555.32

ОБЗОР МЕТОДИК ДИНАМИЧЕСКОЙ ВИЗУАЛИЗАЦИИ В ВЕБ-ПРИЛОЖЕНИЯХ

Фомин А.А.

Научный руководитель – к.т.н., доцент Сокуренько Ю.А.

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

В работе проведен разбор методов динамической визуализации в веб-приложениях, определены особенности динамической визуализации и способов ее реализации.

Ключевые слова: анимация, динамическая визуализация, веб-технологии.

Динамическая визуализация (анимация) существуют уже давно. В настоящее время, когда браузеры и мобильные процессоры становятся все лучше, анимации начинают играть все более важную роль в интерфейсах. Анимации делают интерфейс более понятным, а процесс взаимодействия – более насыщенным. Взаимодействие пользователей с современными веб-сайтами серьезно завязано именно на анимации. Она способна сообщать о неких состояниях страницы, направлять внимание. Анимация помогает пользователю увидеть результат его действий и может влиять на его поведение.

При помощи анимации пользователю проще понять иерархическую структуру интерфейса, поскольку он видит элементы в контексте. Это создает фокус. Пользователь понимает каждую операцию, потому что напрямую видит, как она происходит. Отобразить на экране всю необходимую информацию – это искусство, особенно когда дело касается небольших экранов и плоского дизайна.

В период 2000-х годов все анимированные элементы и эффекты на сайтах были созданы с помощью технологии Flash [1]. На тот момент это была революционная технология, хотя она была очень тяжелой и значительно увеличивала время загрузки сайта. В наши дни анимация создается с использованием более легкого способа кодирования JavaScript и CSS (Cascading Style Sheets), которые помогают добавлять движущиеся элементы на сайт, не перегружая его. Дабы форсировать отказ от технологии Flash, Google объявила о планах по постепенной вычистке Flash из своего браузера Chrome. За последние четыре года использование Flash снизилось на 90%.

Некоторое время в области анимации лидирующие позиции занимал JavaScript (JS). Если говорить об использовании JS для анимации, то можно пойти двумя путями: писать код «с нуля»; либо использовать готовые библиотеки и фреймворки, которых на данный момент имеется огромное количество. Некоторые из них узконаправленные, некоторые имеют широкий спектр применения.

GreenSock Animation Platform или GSAP – это, пожалуй, одна из самых функциональных JavaScript-библиотек для анимирования. Частично эту библиотеку затрагивает Т.В. Павлова [2] в своей работе. В качестве плюсов автор указывает: большое количество методов; легкий синтаксис; синхронизация с частотой обновления экрана. Минусом, с точки зрения автора, является необходимость загрузки и поддержки внешних библиотек. Существует также набор Open Source модульных библиотек и инструментов для анимирования в веб-приложениях. Create.js включает в себя Sound.js, Preload.js и Easel.js библиотеки, которые могут работать как вместе, так и по отдельности. Каждый модуль отвечает за свою часть работы и не пересекается с остальными, поэтому при надобности можно заменить на аналогичный по функционалу от других разработчиков. Но, для тех кто намерен использовать весь набор без изменений, предусмотрена возможность их связывания для упрощения работы и написания кода.

Еще одна библиотека для анимирования с помощью JS – Tween.js, является дополнением к easel.js – одному из самых популярных JavaScript-фреймворков для удобной и легкой работы с Canvas. Тем не менее, tween не имеет никаких зависимостей и может использоваться как вместе, так и отдельно от easel. Основная задача библиотеки tween.js – это анимация в Canvas. И она отлично с ней справляется, делая за вас и упрощая многие вещи.

Move.js – JavaScript-библиотека, позволяющая делать все то же, что и tween.js, только с использованием CSS3 и HTML5 вместо Canvas. Она является не самой легкой, но одной из самых популярных. Набор заранее заготовленных видов анимации (поворот, наклон, ротация и т.д.) делает ее также одной из самых легких в использовании.

По статистике, почти 50% сайтов используют JQuery (JQuery был опубликован на компьютерной конференции «Bar Camp» в Нью-Йорке Джоном Ресигом в 2006 году). В своей книге Д.С. Макфарланд [3] описал основные принципы взаимодействия JavaScript и данной библиотеки. JQuery содержит несколько кросс-браузерных методов для анимации элементов, например, скольжение и плавное исчезновение, без привлечения дополнительных библиотек или плагинов.

Также S. Drasner [4] описывает такую библиотеку как Velocity. По словам автора, она менее производительна чем CSS, но имеет больший потенциал для модификаций разработчиками. Пользовательские модификации могут принести больше функционала в будущем.

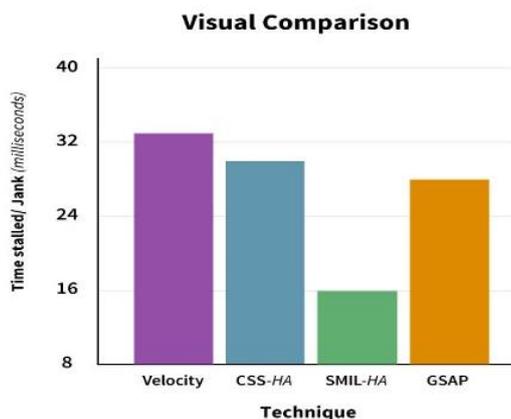


Рисунок. График быстродействия технологий анимирования

В данной работе также проведено исследование быстродействия различных технологий анимирования в веб. Автор создавала анимированные объекты с помощью разных методов (библиотеки `velocity.js`, `CSS`, `SMIL` и `GSAP`) и фиксировала скорость обработки анимации браузером (все тесты запускались на Chrome версии 40.0.2214.91), затем она усреднила все результаты тестов и вывела график скорости обработки. На графике (рисунок) видно, что самой быстродействующей является `SMIL`. В своей другой статье S. Drasner [5] более подробно останавливается на этой технологии. Автор разбирает все преимущества библиотеки анимирования `SMIL`, но также отмечает, что поддержка данной технологии идет на убыль в веб-сообществе. Это означает, что рано или поздно от `SMIL` придется отказаться.

На данный момент появилась альтернатива JavaScript и его фреймворкам – активно разрабатываемая спецификация `CSS3`. Ее разработка еще далека от завершения. Большинство модулей все еще продолжает совершенствоваться и модифицироваться, и ни один браузер не поддерживает все модули. Это означает, что `CSS3` испытывает такие же сложности, как и `HTML5`. Веб-разработчикам нужно решать, какие возможности использовать, а какие игнорировать, а также каким образом заполнить зияющие пробелы в браузерной поддержке. Анимирование с помощью `CSS` затрагивает в своей работе З. Джилленуотер [6]. В качестве минусов данной технологии автор отмечает отсутствие поддержки в старых версиях браузеров, например, Internet Explorer 8.

Спецификация `CSS3` не является частью спецификации `HTML5`. Эти два стандарта были разработаны отдельно друг от друга разными людьми, работающими в разное время в различных местах. Но даже организация W3C призывает веб-разработчиков использовать `HTML5` и `CSS3` вместе, как часть одной новой волны современного веб-дизайна. В `CSS3` появилось много новых свойств, позволяющих создавать анимированные объекты.

Технологию `CSS` и JavaScript сравнил в своей статье J. Doyle [7]. Автор акцентирует свое внимание на том, что у `CSS` и JavaScript нет особых преимуществ друг перед другом. Для разных целей подходят разные методы и технологии. JavaScript имеет множество библиотек и фреймворков позволяющих осуществлять анимирование для разных задач. В свою очередь, консорциум W3C активно совершенствует `CSS`, спецификация `CSS3` тому подтверждение.

Реализация динамической визуализации в сфере веб-технологий занимает все более важную роль. Согласно отечественным и зарубежным исследованиям, посвященным теме динамической визуализации, динамическая визуализация, будучи примененной в сфере веб, решает задачу проектировки гибкого и функционального пользовательского интерфейса. Обеспечивает более комфортное использование веб-приложения конечным пользователем.

Литература

1. Короткевич Д.Э., Воробьев Э.И., Короткевич С.И., Уразова О.А. Flash для WEB-анимации: учебное пособие. – Воронеж: Воронежский государственный технический университет, 2003. – 167 с.
2. Павлова Т.В. Особенности стандартов веб-анимации // Альманах научных работ молодых ученых Университета ИТМО. – 2016. – Т. 3. – С. 88–60.
3. Макфарланд Д. JavaScript и jQuery. Исчерпывающее руководство. – Изд-во: Эксмо, 2015. – 880 с.
4. Drasner S. A Comparison of Animation Technologies [Электронный ресурс]. – Режим доступа: <https://css-tricks.com/comparison-animation-technologies/>, своб.
5. Drasner S. SMIL is dead! Long live SMIL! A Guide to Alternatives to SMIL Features [Электронный ресурс]. – Режим доступа: <https://css-tricks.com/smil-is-dead-long-live-smil-a-guide-to-alternatives-to-smil-features/>, своб.
6. Джилленуотер З. Сила CSS. Освой новейший стандарт веб-разработок! – Питер, 2013. – 304 с.
7. Doyle J. Myth Busting CSS Animations vs. JavaScript [Электронный ресурс]. – Режим доступа: <https://css-tricks.com/myth-busting-css-animations-vs-javascript/>, своб.



Хренкова Марина Андреевна

Год рождения: 1993

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4205

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: khrenkova93@gmail.com

УДК 004.9

ТЕОРЕТИЧЕСКИЙ ОБЗОР OPENSOURCE-ТЕХНОЛОГИЙ ПРОЕКТИРОВАНИЯ ЭЛЕМЕНТОВ ФИРМЕННОГО СТИЛЯ

Хренкова М.А.

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 рассмотрено понятие фирменного стиля, изучены особенности брендбука и гайдлайна, выделены их отличительные черты, проанализированы OpenSource-технологии проектирования элементов фирменного стиля и их коммерческие аналоги.

Ключевые слова: OpenSource, фирменный стиль, брендбук, технологии, программное обеспечение.

В настоящее время фирменный стиль является компонентом корпоративной культуры предприятия и апеллирует к эстетическим ценностям человека. На сегодняшний день существует большое количество специальной литературы, статей, в которых определяется понятие фирменного стиля. Несмотря на это, отдельные компании недооценивают его возможности и не уделяют ему должного внимания. Стоит учитывать то, что правильно подобранный корпоративный стиль может помочь повысить конкурентоспособность, а также уменьшить убытки. Важно с самого начала заложить в восприятии потребителя визуальные ассоциации с предложением компании. В «минимальный комплект» фирменного стиля, по которому клиенты начинают распознавать бренд, обязательно должны входить логотип и фирменный знак, формирующий фирменный блок. Индивидуальный стиль для компании является основополагающей составляющей [1].

В зависимости от задач, которые нужно решить, объем брендбука может быть различным для каждой компании. Одна компания может обойтись 15 страницами, для другой и 80 страниц будет мало. К примеру, брендбук Сбербанка состоит из 250 страниц. Однако важно понимать, что объем брендбука не является целью и ориентироваться на число страниц при его создании не стоит. Самое главное, чтобы решались основные задачи, которые возложены на создание корпоративного стиля [2]. Обычно в брендбук включают паспорт стандартов, который является руководством по работе с фирменным стилем.

Очень часто понятия «брендбук» и «гайдлайн» путают. Брендбук представляет собой совокупность ценностей компании, а также способов донесения их до клиентов. В брендбуке прописываются миссия и философия бренда, его ценности, атрибуты и символика. Он является официальным руководством по управлению брендом.

Гайдлайн, в отличие от брендбука, представляет собой совокупность правил и инструкций по использованию логотипа, знака, персонажа, шрифтовой гарнитуры, цветовых сочетаний.

При проектировании элементов фирменного стиля дизайнер встает перед выбором программного обеспечения (ПО). Для этого нужно вспомнить цель, которая была поставлена изначально – создание фирменного стиля для компании. Следовательно, для создания

элементов брендбука нам понадобится ПО для работы с растровыми и векторными изображениями.

Подобные графические редакторы присутствуют в продуктах таких компаний как Adobe (Photoshop) и Corel (CorelDraw). Среди бесплатного ПО можно выделить GIMP для редактирования растровых изображений и Inkscape для векторных изображений соответственно.

Для того, чтобы проанализировать программы для редактирования растровых изображений, сравним GIMP и Photoshop, чтобы понять, существует ли достойная бесплатная замена коммерческому продукту (табл. 1).

Таблица 1. Сравнение GIMP и Photoshop

Характеристика	GIMP	Adobe Photoshop
Основные инструменты	Почти каждый инструмент имеет свой аналог	Выбор элементов, их выделение, кадрирование, заливка, градиенты, размытие и т.п.
Предпечатная подготовка, управление цветовыми моделями	СМΥК не поддерживается; функции, направленные на подготовку печатной продукции развиты, но слабо	Управлении цветовыми моделями доступно, есть возможность реализовать предпечатную подготовку
Возможность управления текстом	Скудные возможности управления текстом	Возможность создания горизонтального, вертикального текста, текста-маски
Стоимость ПО	Бесплатное ПО	600\$
Ресурсоемкость	Малая ресурсоемкость, инсталлятор Gimp меньше по размеру, поэтому он не занимает на жестком диске много места и быстро устанавливается	Очень ресурсоемкая программа
Чтение форматов других программ	Да. Он также работает и с форматом PSD	Нет

На сегодняшний день GIMP практически не уступает раскрученному бренду Adobe, а по некоторым параметрам, даже превосходит его.

Таким образом, проанализировав коммерческую программу для работы с растровыми изображениями и ее бесплатный аналог, можно говорить о том, что GIMP является достойной альтернативой для разработки фирменного стиля.

Что касается векторных редакторов, то самыми известными проектами среди пользователей являются CorelDraw и его бесплатный аналог Inkscape, который уже около четырех лет разрабатывается сообществом профессионалов. Сравнительный анализ программ представлен в табл. 2.

Таблица 2. Сравнение Inkscape и CorelDraw

Критерий для сравнения	Inkscape	CorelDraw
Способ распространения продукта	GNU GPL	Проприетарный
Стандартные инструменты для создания векторных	Манипулирование предметами, их перемещение; создание примитивов; создание фигур с помощью FreeHand и	Аналогичные инструменты

Критерий для сравнения	Inkscape	CorelDraw
объектов	кривых Безье; градиенты; обводки; перемещающиеся палитры; визуальное редактирование характеристик объектов; другие инструменты для дизайнеров	
Особые возможности	XML-редактор, позволяющий редактировать объекты формата SVG	Возможность допечатной подготовки и работа с базами данных
Операционная система	Кроссплатформенное ПО	Microsoft Windows
Интерфейс	Обводку и палитру цветов необходимо выводить из диалоговых окон	CorelDraw несколько удобнее, так как в нем палитры плавающие
Оптимизация	Используя программу более двух часов в ОС Windows, можно заметить существенное понижение производительности компьютера. Возможны лаги интерфейса, что также усложняет работу	Производительность остается высокой на протяжении всей работы

Отличительной особенностью Inkscape является то, что с его помощью можно создавать документы формата SVG (Scalable Vector Graphics), который позволяет записывать в XML-коде информацию о векторных изображениях, их прозрачности, цвете и других характеристиках [3]. Встроенный в Inkscape XML-редактор дает возможность редактирования параметров, входящих в файл SVG. Древоподобная структура такого документа содержит в себе все объекты, их свойства, слои. Пользователь в любой момент может изменить параметры искомого объекта [4].

Так, любое векторное изображение можно изменить не только с помощью самой графической программы, но и редактируя изображение в структуре древоподобного списка. Подобного не было ни в Adobe Illustrator, ни в CorelDraw.

Еще одна отличительная особенность Inkscape от «мировых гигантов» состоит в точном позиционировании объектов и в их отображении. Любой пользователь вне зависимости от его профессии сможет легко разобраться с интерфейсом векторного редактора. Редактирование векторных объектов в программе Inkscape на сегодняшний день доступно в точном цифровом регулировании параметров моделируемой геометрии [5].

Таким образом, Inkscape также является достойной альтернативой для разработки векторных элементов фирменного стиля.

При создании того или иного вида графики, нужно понимать где она будет применяться. Так как наша работа подразумевает создание элементов корпоративного стиля для компании, то необходимым становится возможность редактирования как растровых, так и векторных изображений. Однако не существует универсального ПО, сочетающего в себе возможности обоих видов графики. Также следует учесть, что для каждого компонента брендбука могут использоваться разные программы. Следовательно, необходимым становится разработка методики комплексного использования бесплатного ПО для создания элементов фирменного стиля.

Литература

1. Бренд-айдентика: рисуем лицо фирмы [Электронный ресурс]. – Режим доступа: <http://ponedelnikmag.com/post/brend-aydentika-risuem-lico-firmy> (дата обращения: 12.01.2018).

2. Ньюман М. 22 непреложных закона рекламы. – М.: АСТ, 2008. – 318 с.
3. Немчанинова Ю.П. Обработка и редактирование векторной графики в Inkscape (ПО для обработки и редактирования векторной графики): учебное пособие. – М., 2008. – 52 с.
4. Бухвостов В.О., Козлов А.А., Ноздря О.Д., Батищев А.В. В поисках оптимального графического редактора // Территория науки. – 2017. – № 2. – С. 91–97.
5. Обработка векторной и растровой графики в свободном ПО [Электронный ресурс]. – Режим доступа: http://vector-raster.narod.ru/sravnenie_vector.html (дата обращения: 12.01.2018).

**Чебыкина Дарья Андреевна**

Год рождения: 1993

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4205Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: chebyk@ro.ru

УДК 615.892**ОСОБЕННОСТИ АВТОМАТИЗАЦИИ ПРОЦЕССА ВЕРСТКИ ПЕЧАТНОГО
ИЗДАНИЯ****Чебыкина Д.А.****Научный руководитель – к.педагог.н., доцент Шуклин Д.А.**

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

В работе рассмотрены проблемы этапа верстки полиграфической продукции. Выявлены основные условия, при которых автоматизация процесса верстки не будет эффективна. Обозначен ряд предпосылок к автоматизации процесса верстки издания. Рассмотрены существующие методы автоматизации верстки.

Ключевые слова: автоматизация, верстка, полиграфия, допечатная подготовка.

Современная полиграфическая технология включает в себя три основных этапа: допечатный, печатный и послепечатный. Допечатный этап объединяет все технологические процессы, связанные с подготовкой к печати издания до изготовления печатной формы включительно. В данной работе речь шла об одном из процессов допечатного этапа – верстке издания. Это наиболее сложный и трудоемкий процесс, требующий обработки огромного количества информации. Автоматизировав данный процесс, можно добиться сокращения времени подготовки издания к печати, поэтому цель работы – изучить проблему автоматизации верстки печатного издания.

Верстка в полиграфическом производстве – это процесс формирования страниц издания путем компоновки графических и текстовых элементов в соответствии с разметкой или макетом.

На рынке широко представлены программные средства, позволяющие усовершенствовать процессы верстки издания. К таким инструментам относятся продукты фирмы Adobe (FrameMaker, InDesign, PageMaker), TeX, QuarkXPress, Ridero и другие. Считается, что у каждого из них своя специализация. Правильный выбор программного продукта для верстки обеспечит высокое качество верстки макета издания (за счет программного инструментария для осуществления допечатной обработки авторского оригинала) и сокращение сроков работы. Это, в свою очередь, способствует слаженности процесса верстки. Однако такой трудоемкий процесс нуждается в современных методах автоматизации для ускорения и упрощения работы верстальщика.

Не каждый этап верстки может быть автоматизирован. Во многом, верстка издания – это субъективная и творческая задача. Дизайнер или верстальщик принимает большинство решений относительно вида, положения и размера элементов на странице. На основе тематических исследований [1, 2] были выявлены условия, при которых автоматизация процесса верстки не будет эффективна:

- высокие требования к художественному оформлению издания;
- наличие большого числа элементов и мелких деталей в макете;
- необходимость ручной обработки исходных данных;

- сложный дизайн издания (основанный на интуитивных знаниях, которые сложно описать на математическом языке);
- значительное изменение макета издания с каждым выпуском.

Исходя из выявленных факторов, можно вывести ряд предпосылок к автоматизации процесса верстки издания:

- макет состоит из множества однотипных страниц;
- процесс верстки имеет большой объем входных данных;
- издание постоянно публикуется и имеет множество выпусков в перспективе;
- в макете присутствует переменная и постоянная информация;
- количество блоков в шаблоне ограничено, их форма четко определена;
- страницы не имеют сложных конструкций;
- в макете издания используется простой и постоянный дизайн;
- внимание сфокусировано на расположении информации и ее виде, а не на ее содержании.

Таким образом, не для каждого вида полиграфической продукции возможно и целесообразно автоматизировать этап верстки. Часто в процессе требуются решения дизайнера, а компоновка информации основывается на его интуитивных знаниях. Однако есть случаи, в которых наблюдаются перечисленные предпосылки к автоматизации. К примеру, это может быть выпуск каталога продукции какой-либо фирмы, рекламный журнал или газета, имеющие постоянный и простой дизайн и др.

Что касается методов автоматизации процесса верстки издания, исследования в этой области ведутся в нескольких направлениях. Во-первых, изучаются возможности автоматизации верстки при внедрении в производство редакционно-издательских систем [3]. Во-вторых, для автоматизации процессов верстки используются подключенные плагины, написанные на языке сценариев и выполняющие конкретные задачи. Разработкой и усовершенствованием данных сценариев занимаются многие исследователи ([4, 5] и др.). В-третьих, ряд исследователей в своих работах занимаются вопросами разработки собственного программного приложения для верстки [1].

Таким образом, в ходе проведенного исследования были выявлены предпосылки к автоматизации процесса верстки издания. На основании обозначенных особенностей верстки, печатная продукция для дальнейшего исследования и разработки методики была ограничена до рекламных газет и журналов. Кроме того, были обозначены основные методы автоматизации верстки. Эти методы будут рассмотрены и проанализированы в дальнейших исследованиях. На основе этого анализа будет разработана авторская методика автоматизации процесса верстки.

Литература

1. Grahn K.J. Catalogue Production Automation – Case Studies [Электронный ресурс]. – Режим доступа: https://www.nada.kth.se/utbildning/grukth/exjobb/rapportlistor/2006/sammanf06/grahn_karl-johan.html, своб.
2. Назарова С.А. Поддержка решений по выбору программно-технических средств верстки издания [Электронный ресурс]. – Режим доступа: <https://trudy.vntu.edu.ua/index.php/trudy/article/view/7/7>, своб.
3. Гончаренко О.Г., Алексеев В.Ф. Инновационный подход в создании издательских систем нового поколения // Актуальные направления научных исследований XXI века: теория и практика. – 2015. – Т. 3. – № 7-4. – С. 21–24.
4. Захарова О.С., Шогин А.Н., Старцева О.Б., Олейникова О.Л. Автоматизация верстки периодических реферативных изданий ВИНТИ РАН в системе AdobeInDesign // Научно-техническая информация. – 2013. – № 5. – С. 31–34.
5. Романовская В.Е., Бизюк А.В., Некрасова Н.Н. Автоматизация процесса верстки рекламных объявлений в газетных изданиях // Технологический аудит и резервы производства. – 2013. – Т. 2. – № 2(10). – С. 3–5.

**Челокьян Лусинэ Арутюновна**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4106Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: lusinechelokian@gmail.com

**Шалобаев Евгений Васильевич**

Год рождения: 1947

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.т.н., доцент

e-mail: kpd@limtu.ru

УДК 004.942

**ПОДГОТОВКА МОДЕЛИ ИЗДЕЛИЯ ДЛЯ ПОСЛЕДУЮЩЕЙ 3D-ПЕЧАТИ
С УЧЕТОМ МЕТОДОВ ИНЖЕНЕРНОГО АНАЛИЗА****Челокьян Л.А.****Научный руководитель – к.т.н., доцент Шалобаев Е.В.**

В работе рассмотрена возможность применения методов компьютерного инженерного анализа с целью адаптации модели для 3D-печати и соответствия будущего изделия определенным требованиям.

Ключевые слова: компьютерный инженерный анализ, адаптация, 3D-печать.

Компьютерный инженерный анализ избавляет от необходимости проведения реальных экспериментов, благодаря возможности симулировать различные физические процессы виртуально. Также, благодаря методам инженерного анализа можно оценить корректность модели для 3D-печати.

Не любая компьютерная модель пригодна для 3D-печати. Подготовка модели изделия для последующей 3D-печати – это комплекс действий по адаптации 3D-модели, который делает ее печать возможной. Существуют определенные требования к параметрам и конструкции модели, которые могут относиться ко всем моделям или зависеть от метода и материала печати [1].

Также требования к модели для печати можно разделить на две группы еще следующим образом:

1. требования, от соблюдения которых зависит возможность и качество печати. Их необходимо всегда учитывать, какое бы изделие не печаталось;
2. требования, от соблюдения которых зависит работоспособность будущего изделия. Они могут быть различными для разных изделий или не применяться вовсе. Для создания корректной модели, соответствующей этим требованиям, необходимо предварительно произвести анализ модели в программах компьютерного инженерного анализа.

Первым из таких требований является минимальная толщина стенок изделия (рис. 1, а). Это требование является обязательным для любой модели, так как поверхность модели должна иметь толщину, чтобы обладать физическим объемом и ее печать была возможна. Минимальная толщина зависит от материала и метода печати, характеристик принтера, и конструкции модели, так как она должна выдерживать свой собственный вес. Когда к будущему изделию предъявляются определенные требования по прочности конструкции,

при выборе толщины стенок стоит произвести анализ прочности модели при тех или иных нагрузках или условиях в программах компьютерного инженерного анализа (CAE) [2].



Рис. 1. Минимальная толщина стенок изделия (а); распределение веса (б)

Следующее требование – это плоское основание модели. Плоское основание позволит модели лучше держаться на столе принтера при печати послойным наплавлением (FDM), если модель печатается, например, (селективным) лазерным спеканием (SLS), этим требованием можно пренебречь. Но помимо возможности печати от этого требования зависит устойчивость будущего изделия. Таким образом, в зависимости от того, будет ли изделие подвергаться каким-либо нагрузкам или должно быть устойчивым в состоянии покоя, следует произвести анализ модели в программах компьютерного инженерного анализа (CAE) на устойчивость или на порог устойчивости при определенных нагрузках соответственно [3].

Правильное распределение веса (рис. 1, б) и минимальное количество нависающих элементов. Для нависающих элементов при послойном наплавлении (FDM) необходимы опоры, которые будут их поддерживать. На печать таких опор уходит рабочее время и материал, а их последующее удаление портит поверхность модели, поэтому их стоит избегать. Также от нависающих элементов зависит правильность распределения веса и устойчивость будущего изделия, как в состоянии покоя, так и при влиянии внешних условий или нагрузок. Компьютерный инженерный анализ также позволяет учесть это требование при конструировании модели и определить возможные условия эксплуатации изделия при допустимой устойчивости с тем или иным количеством нависающих элементов [2, 4].

Полезный объем (рис. 2, а). Для экономии материала и времени печати модели стоит делать полыми. В том случае, когда это невозможно, необходимо выбрать паттерн (рис. 2, б) и процент заполнения модели, который будет соответствовать требованиям прочности напечатанного изделия. Например, изделие с гексагональным заполнением показывает большую прочность, чем с волнообразным. Однако, если изделие должно обладать некоторой гибкостью или скручиваемостью, стоит остановиться на втором варианте. Также на эти параметры влияет процент заполнения модели. Чем он больше (рисунок заполнения становится мельче), тем выше прочность будущего изделия и ниже гибкость. С помощью компьютерного инженерного анализа необходимо подобрать подходящий паттерн и процент заполнения модели, учитывающий требования к прочности, гибкости, скручиваемости и т.д. будущего изделия [5].

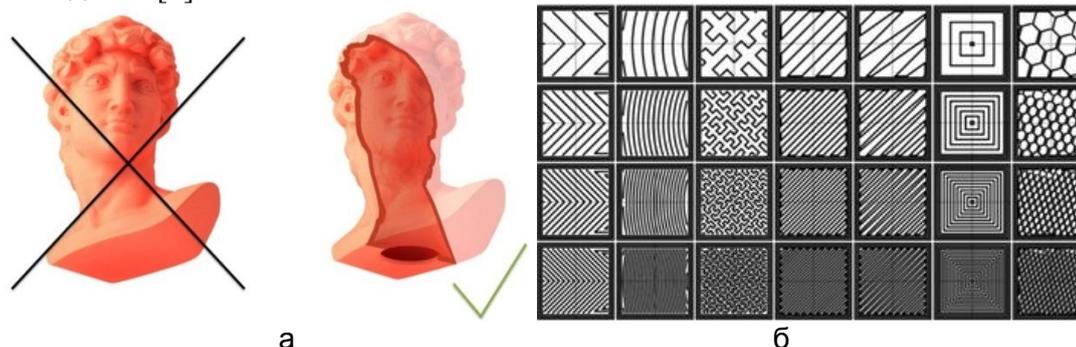


Рис. 2. Полезный объем (а); паттерны заполнения модели (б)

Таким образом, когда будущее изделие должно обладать определенными физико-механическими параметрами и являться работоспособным при определенных условиях эксплуатации, необходимо произвести компьютерный инженерный анализ модели, который позволяет подобрать оптимальные характеристики модели с учетом различных требований.

Литература

1. Медунецкий В.М., Романов Н.А. Компьютерное моделирование и визуализация технологии изготовления нестандартных зубчатых колес из полимерных композиционных материалов // Приборостроение. – 2015. – № 5. – С. 397–400.
2. Правила 3D-моделирования для 3D-печати [Электронный ресурс]. – Режим доступа: <https://www.cubicprints.ru/tutorials/pravila-3d-modelirovaniya-dlya-3d-pechati>, своб.
3. Аддитивные технологии [Электронный ресурс]. – Режим доступа: <http://nami.ru/directions/technical/technology-centre/rapid-prototyping-rapid-prototyping>, своб.
4. Как подготовить модель к 3d-печати [Электронный ресурс]. – Режим доступа: <https://www.m207.ru/sites/default/files/file/Prototypster-3D-Printing-Instruction-Web.pdf>, своб.
5. Основные принципы подготовки 3d-моделей к использованию на 3d-принтере [Электронный ресурс]. – Режим доступа: <https://pyramid3dmc.ru/podgotovka-3d-modeli-dlja-trehmernoj-pechati/>, своб.



Шайхетдинова Алиса Раисовна

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4108

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: alisa.shayk@gmail.com



Шуклин Дмитрий Анатольевич

Год рождения: 1979

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: do@limtu.ru

УДК 004

**ИССЛЕДОВАНИЕ МЕТОДОВ ПОИСКОВОЙ ОПТИМИЗАЦИИ
МОБИЛЬНЫХ ВЕРСИЙ САЙТА**

Шайхетдинова А.Р.

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

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

В данной работе приведен обзор методов для продвижения сайта в поисковых системах, определены особенности мобильного поиска и проведено исследование существующих методов мобильной оптимизации, соответствующих выявленным особенностям.

Ключевые слова: поисковая оптимизация, продвижение, SEO, мобильный поиск, мобильный сайт.

В современном информационном обществе сложно представить конкурентоспособную фирму без своего сайта. Часть бизнеса, в том числе выручка, зависит от позиции этого сайта в поисковой выдаче, поэтому поисковая оптимизация сайтов (SEO, Search Engine Optimization) является важной составляющей продвижения организации.

В SEO существуют различные стратегии для улучшения внутренних, внешних и поведенческих факторов [1].

1. Перелинковка страниц. Ранжирование страницы в значительной степени зависит от ее связи с другими страницами Web.
2. Устранение дубликатов. Из-за дубликатов размывается ссылочный вес, а контент теряет уникальность.
3. Инструкции для поисковых систем. С помощью них можно настроить индексацию сайта.
4. Работа с авторским контентом.

Поисковые системы (ПС) регулярно совершенствуют алгоритмы ранжирования. Это необходимо для более релевантной выдачи по поисковому запросу. Важным толчком к последнему глобальному изменению алгоритмов стало преобладающее использование мобильных устройств для поиска. Для мобильного пользователя стало важно не только релевантное содержание страницы, но и удобство использования ресурса. Отсутствие мобильной адаптации может стать причиной низкого трафика и потери клиентов, которые с большой долей вероятности приобретут товар на мобильной версии сайта конкурента.

Особенности мобильного поиска:

- отсутствие физической клавиатуры и маленький размер экрана. По этой причине пользователи формируют более короткие запросы, содержащие меньшее число ключевых слов. Исследование этой особенности было проведено в работе [2];
- геолокация. Необходимо использовать это преимущество при реализации мобильного приложения;
- маленькое окно ПС. Пользователь одновременно видит первые 2–4 позиции выдачи. Из этого есть два важных следствия: попасть в топ выдачи сложнее и необходимо большее внимание уделять сниппетам;
- маленький размер экрана. Важен размер шрифта в заголовках, описании сайта. Вся структура должна быть рассчитана для удобства чтения на устройствах с диагональю 4,7–5,5 дюймов;
- голосовой поиск. Важна оптимизация под голосовые запросы;
- ограниченное время на поиск. Необходимо настроить качественный поиск по сайту и обеспечить высокую скорость загрузки страниц.

Мобильные пользователи предпочитают изучать и получать информацию с mobile-friendly web-сайтов, поэтому эти страницы находятся в топе выдачи по поисковым запросам. Ниже приведены методы, позволяющие сделать сайт mobile-friendly [3].

- Использование адаптивного дизайна. Анализ эффективности использования адаптивного дизайна проведен в работе [4]. Чтобы сообщить браузеру, что страница адаптируется к любым устройствам, нужно добавить в заголовок документа мета-тег: `meta name="viewport" content="width=device-width, initial-scale=1.0"`.
- Настройка видимость JS/CSS файлов. В случае, если эти элементы заблокированы, сайт не пройдет тест ПС на проверку соответствия сайта принципу mobile-friendly.
- Воспроизводимость контента в мобильной версии сайта. Во встраиваемых видео рекомендуется использовать форматы, которые поддерживаются на всех устройствах, а также добавлять расшифровку видео.
- Ускорение загрузки. Скорость ответа – один из главных факторов успеха web-сайта.
- Оптимизация дизайна. Мобильная версия сайта должны быть приятной для восприятия и удобной для просмотра. Реализация дизайна для большого пальца.
- Отказ от Flash и pop-up окон. Flash не поддерживается большинством современных мобильных устройств, а pop-up окна снижают удобство просмотра.
- Соответствие мобильного URL десктопному эквиваленту. Если сайт имеет отдельные URL для разных версий сайта, необходимо обеспечить их соответствие во избежание проблем с ранжированием (удаление одной из версий из индексации в связи с дублированием контента).

Мобильные устройства развиваются, число мобильных пользователей растет вместе с долей мобильного поиска, а также появляются новые особенности. В дальнейшем планируется усовершенствование методов оптимизации мобильного сайта ввиду новых факторов.

Литература

1. Lemos J.Y., Joshi A.R. Search engine optimization to enhance user interaction // IEEE, IoT in Social, Mobile, Analytics and Cloud (I-SMAC). – 2017. – P. 398–402.
2. Hattori S., Tezuka T., Tanaka K. Context-Aware Query Refinement for Mobile Web Search // Applications and the Internet Workshops – 2007. – 15 p.
3. Schubert D. Influence of Mobile-friendly Design to Search Results on Google Search // 19th International Conference Enterprise and Competitive Environment. – 2016. – P. 424–433.
4. Jiang W., Zhang M. Responsive web design mode and application // Advanced Research and Technology in Industry Applications (WARTIA). – 2014. – P. 1303–1306.



Шакирзянова Евгения Ильдаровна

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4205

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: jenya.1994@inbox.ru

УДК 004

**СОЦИАЛЬНЫЕ СЕРВИСЫ КАК ИНСТРУМЕНТ SMM-ПРОДВИЖЕНИЯ
ОБРАЗОВАТЕЛЬНЫХ УСЛУГ**

Шакирзянова Е.И.

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

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

В работе исследованы основные задачи и возможности SMM-продвижения образовательных услуг, приведены способы эффективного взаимодействия с аудиторией, описаны тенденции, преобладающие в процессе продвижения, выделены площадки для осуществления SMM-деятельности в образовательных процессах России.

Ключевые слова: социальные сети, SMM-продвижение, образование, маркетинг, сервисы, технологии, информация.

В России 90% интернет-пользователей имеют зарегистрированный в социальной сети аккаунт [1]. Социальные сети – площадки, созданные пользователями для пользователей, для обмена сообщениями, информацией, опытом, для продвижения товаров, услуг, в том числе и образовательных. Продвижение образовательных услуг подразумевает создание позитивного имиджа компании (организации) или физического лица, поэтому его осуществление происходит посредством использования средств и инструментов, не имеющих финансовой или коммерческой основы. Продвижение образовательных услуг при помощи социальных сетей и сервисов называется SMM-продвижением (Social Media Marketing) [2].

SMM образовательных услуг – комплекс работ, включающий контроль и решение следующих задач:

- привлечение пользовательской аудитории;
- выявление проблемных областей, актуальных для пользователей;
- создание качественного контента, ориентированного на пользователей;
- информационное оповещение аудитории о новых образовательных услугах или направлениях деятельности вуза, увеличение известности бренда;
- повышение заинтересованности пользователей;
- установление обратной связи с аудиторией;
- увеличение посещаемости ВУЗовских (факультетских, кафедральных) порталов.

Решение указанных задач осуществляется посредством прямого общения с пользователями через социальные сервисы и сети, выступающие в роли каналов связи. При этом необходимо понимать тематическую направленность каждой из социальных сетей, преобладающую аудиторию в них, используемый инструментарий для построения дальнейшей успешной работы с аудиторией.

Для эффективного взаимодействия с пользователями в SMM существуют следующие способы:

- видеоролики (называемые также вирусные ролики), основная цель которых за короткое время раскрыть идею образовательной услуги, продемонстрировать ее в действии (отрывок урока);
- ведение блогов как личных, так и блогов образовательных организаций, представляющих собой собрание полезных рекомендаций;
- тематическая разбивка информации, обеспечивающая максимально понятную работу с контентом;
- интересный контент, представленный разнообразными форматами (графика как анимированная, так и статичная, аудиозаписи, видео), – один из залогов успешного продвижения;
- наличие ссылок на соответствующие тематике ресурса источники, а также ссылки, находящиеся на внешних источниках на ресурс организации, иначе говоря, рекомендации извне [2].

Успешность продвижения образовательных ресурсов в социальных сетях определяется положительным увеличением качественных показателей, рост которых напрямую зависит от знания трендов социальных сетей. Ежедневное развитие сервисов, постоянный рост изменений позволяет отслеживать преобладающие в процессе продвижения тенденции.

- Stream-трансляции, иначе говоря, прямые эфиры, где посредством онлайн-вещания освещается текущая деятельность организации [3]. Поточковые видео доступны в таких широко используемых сетях в России, как Instagram, ВКонтакте, Facebook Live и Twitter. Трансляции могут стать регулярными, что обеспечит наибольшее привлечение аудитории. При выборе платформы для онлайн-вещания необходимо отдать предпочтение сервису, где число потенциальных пользователей максимальное.
- Исчезающий контент. Обилие информации в последнее время влечет за собой безразличное отношение пользователей к потребляемому и просматриваемому ими контенту. Исчезающий контент впервые задан социальным сервисом Snapchat и направлен на искусственное повышение ценности контента путем его быстрого удаления. Данное течение вскоре подхватили социальные сети Instagram и ВКонтакте. Короткие видео пользуются большей популярностью, так как не отнимают много времени при просмотре, предоставляют дозированную информацию, содержат только факты.
- Таргетинг аудитории. Правильная настройка последнего позволяет показывать каждому пользователю только те посты, которые могут быть ему наиболее тематически интересны [4].
- Качественный контент. Контент – один из самых важных инструментов в продвижении любых услуг. В сфере образования данному фактору необходимо уделять наибольшее внимание. Важно не допустить ошибку, сделать материал интересным, удобочитаемым, понятным, доступным, внешне привлекательным, т.е. снабдить его видео или графическим сопровождением. Контент должен представлять информационную ценность, задерживать внимание пользователей, быть полезным и исчерпывающим.
- Мессенджеры и чат-боты. Основная задача – быстрая консультация, помощь и поддержка клиента на ресурсе. Удобные в использовании, позволяют популяризировать услуги путем нестандартного их представления и многоканального распространения [1].
- Личный подход. Актуальность персонализации объясняется желанием каждого пользователя стать «особенным» и получать только ту информацию, которая ему интересна и необходима.
- Авторские рекомендации. Экспертное мнение преподавателей, советы научных работников и заведующих отделениями образовательных организаций представляют собой ценную информацию. Зачастую выбор ресурса пользователями (студенты, абитуриенты, желающие начать образовательный процесс) среди огромного множества имеющихся зависит от уровня его рекомендательной составляющей. Чем она выше (чем больше ссылаются на ресурс и чем больше цитируют его заслуженные деятели

образования), тем больше пользовательского доверия, соответственно, выше шансы образовательного ресурса быть выбранным аудиторией.

- VR-технологии. Использование технологий виртуальной реальности позволяет пользователям глубже проникнуться сутью ресурса. К тому же, благодаря им, любой контент становится наиболее зрелищным, что увеличивает интерес аудитории.

С каждым днем социальные сети становятся все более ориентированными на конкретного пользователя. Это преимущество позволяет образовательным учреждениям повышать коэффициент заинтересованности аудитории в содержательной части ресурса, увеличивать пользовательскую отдачу и продвижение услуг обучения.

Анализ социальных сетей, популярных в России, их системное описание и структурный разбор инструментария и технологических возможностей позволяют выделить в качестве основных площадок для работы в области SMM-продвижения образовательных услуг социальную сеть ВКонтакте, в связи с ориентированностью на российского пользователя, сервис Telegram, в связи с обеспечением новых и нестандартных подходов к представлению информации пользователям, также в качестве вспомогательных площадок социальные сети Одноклассники (для максимального распространения контента среди пользователей России и СНГ), Facebook (для реализации образовательных услуг на международном уровне), Instagram (для срочных объявлений), Youtube – в качестве носителя обучающего и просветительского видеоконтента организации. Совместная работа всех сервисов обеспечит максимально результативный исход SMM-деятельности в сфере образовательных услуг, повысит ключевые показатели эффективности, выведет услуги обучения на новый уровень [5, 6].

Литература

1. Прохоров А.В., Семишова Е.П. Современные технологии продвижения образовательных услуг // Вестник Тамбовского Университета. – 2014. – № 10(138). – С. 47–51.
2. Халилов Д. Маркетинг в социальных сетях. – М.: Манн, Иванов и Фербер, 2013. – 201 с.
3. Воробьева А.М. Методы продвижения образовательных услуг с помощью образовательного портала // Вопросы современной науки и практики. – 2011. – № 3(34). – С. 96–102.
4. Нетология – университет интернет-профессий [Электронный ресурс]. – Режим доступа: <https://netology.ru/blog/vybor-smm> (дата обращения: 17.01.2018).
5. Ingate [Электронный ресурс]. – Режим доступа: http://www.freelancers.net/posted_files/NE3045F2DA2F7.pdf (дата обращения: 21.09.2017).
6. Фещенко А.В. Социальные сети в образовании: анализ опыта и перспективы развития. [Электронный ресурс]. – Режим доступа: https://ido.tsu.ru/files/pub2011/19_Feschenko.pdf (дата обращения: 15.11.2017).

**Швалева Дарья Сергеевна**

Год рождения: 1995

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4107Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: dashka.shvaleva@mail.ru

**Государев Илья Борисович**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, к.педагог.н., доцент

e-mail: goss@corp.ifmo.ru

**Сокуренок Юрий Андреевич**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», к.т.н., доцент

e-mail: kpd@limtu.ru

УДК 004.4**АНАЛИЗ ЗАРУБЕЖНЫХ И ОТЕЧЕСТВЕННЫХ ИСТОЧНИКОВ ПО ТЕМЕ
«СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОЛОГИЙ УПРАВЛЕНИЯ ОБЪЕКТНОЙ
МОДЕЛЬЮ ВЕБ-ДОКУМЕНТА»****Швалева Д.С., Сокуренок Ю.А., Государев И.Б.****Научный руководитель – к.т.н., доцент Сокуренок Ю.А.**

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

В работе рассмотрен анализ методологий управления объектной моделью веб-документа. Представлены преимущества и недостатки методологий манипулирования DOM. Сделаны выводы относительно исследуемых методологий управления объектной моделью веб-документа.

Ключевые слова: методология, объектная модель документа, Document Object Model, DOM, JavaScript, фреймворк.

Каждодневное развитие web-технологий и пользовательских интерфейсов актуализировало проблему получения доступа к содержимому HTML, XHTML и XML-документов, а также изменения содержимого, структуры и оформления таких документов.

Все необходимые файлы для web-страницы передаются в web-браузер, где они транслируются из текстовой в объектную модель. Это актуально как для простого кода, так и для сложного, расположенного в одном файле или в отдельных файлах. Браузер напрямую работает с объектной моделью, а не с предоставленными текстовыми файлами. Используемая браузером модель называется Document Object Model (DOM).

Document Object Model – это инструмент, с помощью которого JavaScript видит содержимое HTML-страницы и состояние браузера. Он не является языком программирования, но без JavaScript он не имеет никакой модели или представления о веб-странице, HTML-документе, XML-документе и их элементах.

В настоящее время веб-разработчики могут выбирать из широкого круга библиотек и фреймворков JavaScript, обеспечивающих возможность манипуляции DOM, но сложно остановиться на одной конкретной [1]. Это связано с большим количеством различных методологий управления объектной моделью веб-документа, которые решают большинство задач разработки и имеют свои преимущества и недостатки.

С помощью Document Object Model можно создавать документы, перемещаться по их структуре и добавлять, изменять или удалять элементы и содержимое [2]. Все, что содержится в документе HTML или XML, может быть доступно с помощью DOM. Любой документ с помощью DOM может быть представлен в виде дерева узлов, каждый узел которого представляет собой элемент, атрибут, текстовый, графический или любой другой объект. По существу DOM соединяет веб-страницу с языками описания сценариев, либо с языками программирования.

Результаты анализа отечественных и зарубежных исследований показали, что DOM является не зависящей от языка спецификацией; другими словами, есть возможность использовать ее во многих популярных языках программирования. Консорциум Всемирной паутины (World Wide Web Consortium, W3C) определяет для DOM несколько правил связывания с разными языками. Правила связывания языка – это просто API, определенный для разрешения использования DOM в конкретном языке. Например, вы можете найти однозначно определенные правила связывания DOM для C, Java и JavaScript. Следовательно, вы можете использовать DOM в любом из этих языков.

Чаще всего разработчики используют язык сценариев JavaScript. Простота создания динамических элементов пользовательского интерфейса стала играть ключевую роль в веб-разработке, прославила JavaScript. С этим связано быстрое появление различных библиотек и фреймворков JavaScript.

Библиотека React была одна из первых, которая реализует виртуальное DOM-дерево. Виртуальный DOM представляет легковесную копию обычного DOM. И отличительной особенностью React является то, что данная библиотека работает именно с виртуальным DOM, а не обычным [3]. Такая схема взаимодействия с элементами веб-страницы работает гораздо быстрее и эффективнее, чем если бы мы работали из JavaScript с DOM напрямую.

Dojo – библиотека, очень схожая с jQuery. Она – содержит все необходимые функции для манипуляции с DOM, хелперы событий и анимации, также AJAX-функции. Главной особенностью Dojo является простота использования и многофункциональность. С помощью данной библиотеки можно работать с DOM структурой HTML-документа, а именно: находить элементы, добавлять, модифицировать [4].

В 2016 году большой рост популярности получил фреймворк Vue.js. Vue – один из самых быстроразвивающихся JS-фреймворков в 2016 г. Vue описывает себя как «Интуитивный, Быстрый и Интегрируемый MVVM для создания интерактивных интерфейсов».

Одним из ключевых моментов в работе Vue.js является виртуальный DOM. Он так же, как и React представляет легковесную копию обычного DOM.

Одной из самых популярных JS-библиотекой всех времен является jQuery [5].

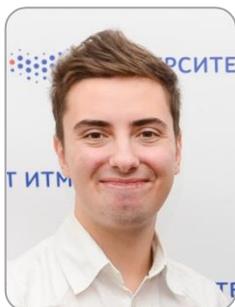
Она помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Она является одной из лучших библиотек, если необходимо управлять DOM или применять AJAX-запросы.

Согласно анализу, разработчики уделяют достаточно большое внимание выбору методологии управления объектной моделью документа. Объясняется это тем, что манипулирование DOM, будучи применимым в веб-разработке, решает важные задачи. Для

каждого разработчика важна стабильность, простота понимания написанного кода, а также скорость обработки и написания кода, а различные библиотеки и фреймворки решают это вопросы по-разному. Рассмотрены одни из самых популярных и используемых инструментов манипулирования объектной моделью веб-документа. Тема общего назначения – манипуляции объектной моделью. Выделены их преимущества и недостатки. Вопрос сравнения практического применения осящен недостаточно детально. Были выделены моменты, связанные с работой Document Object Model и его управлением, которые требуют более детального изучения как с точки зрения теории, так и с точки зрения сравнения практического применения методологий управления объектной моделью веб-документа. Исходя из всего вышесказанного, представляется необходимым проводить дальнейшие исследования по данной теме.

Литература

1. Медведев Ю.С., Терехов В.В. Достижение максимальной производительности AJAX-приложений // Вестник АГУ. – 2013. – Вып. 4. – С. 121–124.
2. Document Object Model (DOM) Level 1 Specification [Электронный ресурс]. – Режим доступа: <https://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/DOM.pdf> (дата обращения 03.10.2017).
3. Карышев А.А., Афанасьев В.Р. Разработка web-приложения для автоматизированной генерации документов на основе docx-шаблонов // Изв. ТулГУ. – 2017. – Вып. 5. – С. 294–296.
4. Матвеев А.Г., Якубайлик О.Э. Разработка веб-приложения для обработки и представления пространственных метаданных геопортала // Сибирский журнал науки и технологий. – 2012. – С. 49–54.
5. Chaffer J., Swedberg K. JQuery Reference Guide [Электронный ресурс]. – Режим доступа: <https://books.google.ru/books?hl=ru&lr=&id=JpEQJmidAbUC&oi=fnd&pg=PT35&dq=jQuery&ots=XkqFnYkvj-#v=onepage&q=jQuery&f=false> (дата обращения: 07.11.2017).



Щербаков Александр Борисович

Год рождения: 1993

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: aleksandersher@gmail.com



Готская Ирина Борисовна

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, д.педагог.н., профессор

e-mail: iringot@mail.ru

УДК 004.9

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ СЕРВЕРНЫХ ФРЕЙМВОРКОВ НА ПЛАТФОРМЕ
NODE.JS**

Щербаков А.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 проведен сравнительный анализ серверных фреймворков на платформе Node.js. В качестве критериев оценки использовались измерения качества исходного кода фреймворков с помощью метрик программного обеспечения, валидности исходного кода, поддерживаемости и производительности.

Ключевые слова: веб-программирование, веб-приложения, JavaScript, фреймворки, метрики программного обеспечения.

JavaScript является языком программирования для клиентской веб-разработки. Он также стал популярным среди разработчиков веб-серверов. Одной из основных причин его растущей популярности является создание и развитие Node.js. Node.js является платформой для разработки сервисов и веб-приложений. Фактически, это позволяет разработчикам создавать серверные приложения на JavaScript. Таким образом, разработчику не нужно использовать два разных языка для создания клиентской и серверной части [1–4].

В данной работе был проведен сравнительный анализ трех серверных фреймворков для Node.js: Express.js, Koa.js и Sails.js. Express.js и Koa.js используют подход промежуточного программного обеспечения (ПО). Sails.js использует подход MVC. Выбор основан на информации, расположенной на официальных Github-репозиториях данных продуктов (табл. 1). Ключевым параметром является количество «звезд», которые отражают количество благодарностей, выраженных разработчикам.

Таблица 1. Список серверных фреймворков

Название	Количество «звезд» на github.com	Ссылка на репозиторий
Express.js	35 976	https://github.com/expressjs/express
Koa.js	19 078	https://github.com/koajs/koa
Sails.js	18 230	https://github.com/balderdashy/sails

В данном исследовании проведены измерения и оценки качества исходного кода фреймворков с помощью метрик ПО, валидности исходного кода (отсутствие ошибок и предупреждений выполнения), поддерживаемости и производительности.

Метрики были вычислены при помощи утилит plato (github.com/es-analysis/plato) и complex-report (github.com/philbooth/complexity-report). Тесты валидации были проведены при помощи инструмента ESLint (<https://github.com/eslint/eslint>). Результаты подсчета метрик выбранных фреймворков представлены в табл. 2.

Таблица 2. Показатели качества кода серверных фреймворков

Метрика ПО	Express.js	Koa.js	Sails.js
Количество строк кода	4 050	1 663	13 623
Количество выражений	1876	950	6721
Количество строк комментариев	212	50	1654
Соотношение количества строк комментариев и строк кода	0,05	0,03	0,12
Цикломатическая сложность	366	125	1427
МП	65,67	73,33	68,35
Глубина вложенности	4	3	6
Количество ошибок	21	7	45
Количество предупреждений	274	178	37

Из результатов значений метрик были получены следующие выводы. Sails.js содержит больше всего комментариев в исходном коде, однако имеет самую высокую цикломатическую сложность, глубину вложенности и количество критических ошибок. В то время как Koa.js содержит в своем коде меньше всего комментариев, но имеет самые низкие показатели цикломатической сложности и количества ошибок и самый высокое значение индекса поддержки кода. Express.js содержит больше всего ошибок листинга кода.

Показатели поддержки фреймворков демонстрируют насколько велико его сообщество и насколько хорошо ведется разработка и сопровождение пакета его разработчиками (табл. 3).

Таблица 3. Показатели поддержки серверных фреймворков

Показатель	Express.js	Koa.js	Sails.js
Лицензия	MIT	MIT	MIT
Дата создания	декабрь 2010	ноябрь 2013	январь 2013
Всего версий	258	67	189
Частота выпуска версий	каждые 10 дней	каждые 23 дня	каждые 10 дней
Количество разработчиков	4	5	4
Число зависимостей	30	24	59
Количество скачиваний в день	279 808	16 701	2 642
Количество скачиваний в неделю	3 415 623	88 002	19 132
Количество скачиваний в месяц	14 875 063	340 977	78 555
Открытые вопросы	160	44	63
Открытые запросы на слияние	30	15	18
Количество звезд на github	35 976	19 078	18 230
Количество подписчиков на github	1 781	819	823
Количество форков	6 480	1 700	1 750
Наличие Вики	Да	да	нет

В результате сравнения Express.js является более популярным, был создан раньше, и также имеет большее количество ежедневных, еженедельных и ежемесячных загрузок, подписчиков на Github и форков. У Koa.js самая большая команда разработчиков, меньше

зависимостей, меньше открытых вопросов и запросов на слияние. Sails.js имеет более частые обновления.

Для измерения производительности использовалась утилита Apache Bench, которая посылала GET-запрос на корневую конечную точку для каждого тестового приложения. Тестовые приложения, написанные на трех рассматриваемых фреймворках, реализовывали идентичную функциональность: посылали в ответ простую строку «Hello World!». Необходимо обратить внимание, что данная методика тестирования не обязательно соотносится с поведением в реальном мире, а скорее проверяет теоретическое максимальное количество запросов, которые могут быть обработаны.

Apache Bench была настроена на одновременное выполнение 100 запросов до тех пор, пока не завершится 50 000 запросов или не истечет 20 с. Затем записывались запросы в секунду. Как Apache Bench, так и сервер были размещены в одной и той же среде, удаляются любые факторы, которые могут привести к нестабильности сети. Результаты тестирования производительности представлены в табл. 4.

Таблица 4. Показатели производительности серверных фреймворков

Фреймворк	Express.js	Koa.js	Sails.js
Среднее количество ответов в секунду	1 745	1 887	1 554

Из результатов проведения тестирования производительности были получены следующие выводы: наилучшие показатели производительности демонстрирует Koa.js, в то время как Sails.js является самым медленным из рассматриваемых фреймворков. Данные результаты напрямую коррелируют с показателями метрик ПО, рассчитанных ранее.

Попробуем сделать общий вывод из того, что уже известно о трех рассматриваемых фреймворках. Определим, для каких проектов в целом лучше подходит каждый из них.

Express.js подходит для:

- больших проектов, предполагающих кастомизацию;
- случаев, когда необходима долгосрочная поддержка приложения.

Koa.js подходит для:

- как небольших, так и средних проектов, которые предполагают развитие в перспективе;
- для проектов с высокой степенью кастомизации;
- для облегчения долгосрочной поддержки приложений.

Sails.js подходит для:

- быстрого старта проекта;
- приложений реального времени, где требуется моментальная реакция;
- приложений, не требующих долгосрочной поддержки.

Литература

1. Gizas A., Christodoulou S., Papatheodorou T. Comparative Evaluation of JavaScript Frameworks // Proceedings of the 21st Annual Conference on World Wide Web Companion. – 2012. – P. 513–514.
2. Graziotin D., Abrahamsson P. Making Sense out of a Jungle of JavaScript Frameworks, Towards a Practitioner-friendly Comparative Analysis // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). – 2014. – P. 334–337.
3. How to choose a framework [Электронный ресурс]. – Режим доступа: <https://hackernoon.com/how-to-choose-a-framework-ea8b5b1e1f44> (дата обращения: 25.10.2017).
4. Compare NPM packages [Электронный ресурс]. – Режим доступа: <https://npmcompare.com> (дата обращения: 16.11.2017).

**Яковлева Маргарита Александровна**

Год рождения: 1994

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4207Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: naruyaoi@yandex.ru

**Готская Ирина Борисовна**Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, д.педагог.н., профессор

e-mail: iringot@mail.ru

УДК 004

ОБЗОР МЕТОДОВ АВТОМАТИЧЕСКОЙ МУЗЫКАЛЬНОЙ ТРАНСКРИПЦИИ**Яковлева М.А.****Научный руководитель – к.педагог.н., профессор Готская И.Б.**

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

В работе рассмотрены основные задачи автоматической музыкальной транскрипции, приведен ряд методов и алгоритмов, использующийся для решения выделенных задач. Проведено их сравнение и предложен вариант решения задачи автоматической музыкальной транскрипции в виде представленного набора алгоритмов.

Ключевые слова: автоматическая музыкальная транскрипция, аудио, формат MIDI, ноты.

В настоящее время существует множество задач, связанных с анализом музыкальных аудиозаписей: классификация музыки по жанрам, определение названия композиции по ее отрывку, определение звучащих в композиции музыкальных инструментов, поиск данных о композиции при наличии безымянного аудиофайла [1].

Эти задачи за время исследования музыки научным сообществом получили ряд решений, часть которого характеризуется достаточно высокой точностью, чтобы признать дальнейшие исследования необязательными. Однако в задаче научного исследования музыки присутствует направление, которое, несмотря на множество предложенных решений, ввиду своей сложности, до сих пор не исчерпало себя. Это направление включает себя распознавание голоса и отдельных музыкальных инструментов в музыкальной аудиозаписи, а также преобразование аудиофайла в ноты.

Системы автоматического музыкального преобразования направлены как раз на решение задачи преобразования аудиофайла в ноты. Если рассматривать систему автоматической музыкальной транскрипции (АМТ) как функцию, то на входе получаем аудиосигнал – функцию зависимости амплитуды от времени. На выходе должны получить запись в формате цифрового интерфейса музыкальных инструментов (MIDI). Партитура MIDI представляет собой своеобразный график с высотой нот на вертикальной оси и временем на горизонтальной оси [2]. Используя протокол MIDI, можно сохранить данные об определенной высоте, начале, затухании и громкости нот, а также дополнительные

параметры, такие как тип инструмента, нотный ключ и темп. Хотя существует множество протоколов, используемых для компьютерной нотной записи, таких как MusicXML1 или Lilypond2, автоматические системы транскрипции, предлагаемые в литературе, обычно конвертируют входную запись в MIDI-файл или MIDI-подобное представление (возвращая высоту ноты, ее начало и окончание). Именно этот формат в данное время наиболее распространен и поддерживается различными программами для работы с музыкой, и поэтому рекомендуется к использованию.

Итак, автоматическое музыкальное преобразование (Automatic music transcription, АМТ) – это процесс конвертации акустического музыкального сигнала в символьное представление музыки, такое как файлы формата MIDI, которые содержат информацию о нотах, начале и окончания звучания каждой ноты и могут содержать скорость игры произведения или музыкальный инструмент.

Задачу АМТ можно разделить на следующие шаги:

- построение зависимости амплитуды сигнала от времени;
 - определение точек начала и окончания звучания нот;
 - определение высоты нот.
1. Построение зависимости амплитуды сигнала от времени. Если разделить все подходы к обнаружению нескольких тонов и транскрипции полифонической музыки в соответствии с используемым темпоральным представлением частоты, то мы увидим, что большинство подходов используют краткосрочное преобразование Фурье (STFT) в качестве основного [3]. В других подходах используются методы фильтров, такие как фильтр эквивалентной прямоугольной полосы пропускания (ERB), преобразование константы-Q (CQT) и вейвлет-преобразование. Однако, несмотря на хорошие результаты применения в АМТ преобразования Фурье, основная причина его преобладания в научных работах в том, что оно начала использоваться в данной области еще в 1950-х годах, в то время как вейвлет преобразование и преобразование Q-константы являются более современными методиками. И в сравнение с фурье-преобразованием они дают более точные результаты.
 2. Определение точек начала и окончания звучания нот. НМР – неотрицательное матричное разложение;
 - неотрицательная обратная матричная свертка;
 - PLCA – вероятностный латентный компонентный анализ;
 - SI-PLCA – инвариантный относительно сдвига вероятностный латентный компонентный анализ;
 - TC-CPM – ограниченная во времени вероятностная модель свертки сигнала;
 - неотрицательные скрытые марковские модели.

Вероятностный латентный компонентный анализ (PLCA) представляет собой прямое расширение вероятностного латентного семантического анализа (PLSI) [4], которое имеет дело с произвольным числом измерений и может проявлять различные характеристики, такие как разреженность или сдвиговая инвариантность.

3. Определение высоты нот. Все методы оценки высоты тона можно подразделить на два основных типа: спектральные и темпоральные. Определить высоту звука путем изучения его спектра можно благодаря тому, что обертоны у гармонических звуков возникают на целых значениях частот, кратных основной частоте этого звука.

Также можно определить высоту одиночного тона в пределах ограниченной временной области. Самый простой подход – использование функции автокорреляции с использованием входного сигнала. Для периодической формы сигнала первый пик в функции автокорреляции указывает на основной период. Однако следует отметить, что пики также встречаются в кратных периоду значениях (также называемых субгармоническими ошибками). В литературе предложено несколько вариантов и расширений автокорреляционной функции, таких как функция разности средних величин, которая вычисляет расстояние между блоками: между блоком сигнала и блоком, сдвинутым на v .

Другим вариантом является функция квадрата-разности, которая заменила расстояние между блоками эвклидовым расстоянием.

Отмечено, что спектральные методы оценки высоты тона имеют тенденцию к ошибкам, появляющимся в значениях, кратных основной частоте (гармоническим ошибкам), в то время как темпоральные методы оценки высоты тона обычно дают ошибки в подмножествах F_0 (субгармонические ошибки). Исходя из этого, было выдвинуто утверждение, что компромисс между спектральными и темпоральными методами потенциально может улучшить точность оценки высоты тона.

Одним из наиболее эффективных спектротемпоральных методов является модель фильтра эквивалентной прямоугольной полосы пропускания (ERB). Эта модель имеет связи с человеческими слуховыми моделями. Унитарная модель состоит из следующих этапов:

1. входной сигнал разделяется блоком фильтров по заданным полосам частот;
2. выход каждого фильтра выпрямляется наполовину;
3. каждый канал проходит сжатие и фильтрацию нижних частот.

Именно спектротемпоральные методы дают наилучшие результаты при обработке монофонической музыки – т.е. музыки, в которой в каждый момент времени звучит только одна нота. Однако при обработке полифонической музыки наилучшие результаты были выявлены у спектральных методов. В большинстве методов оценки высоты нескольких нот используется вариант спектрального метода. Даже система Толонен, которая зависит от сводной функции автокорреляции, использует БПФ для вычислительной эффективности.

Выбранные алгоритмы. Было проведено их сравнение и на основе полученных данных сделан следующий выбор:

1. для создания спектрограмм использовать:
 - непрерывное вейвлет-преобразование;
 - преобразование Q-константы;
2. для определения точек начала нот:
 - спектральный метод определение порогового значения;
3. для определения высоты ноты:
 - SI-PLCA – инвариантный относительно сдвига вероятностный латентный компонентный анализ;
 - TC-CPM – ограниченная во времени вероятностная модель свертки сигнала.

Проведенный анализ показал, что в настоящее время среди продуктов по разработке приложений представлено большое количество кроссплатформенных инструментов, позволяющих снизить затраты ресурсов на создание программного обеспечения. Следовательно, область остается актуальной в настоящее время [5–19].

Литература

1. Новиков А.О. Распознавание нот в аудиофайле // International Scientific Conference «Advanced Information Technologies and Scientific Computing». – 2014. – С. 133–136.
2. Личман Е.Ю. Элементарная теория музыки и основы творческого музицирования. Учебно-методическое пособие. – Павлодар: ПГПИ, 2010. – 48 с.
3. Marolt M. Фурье и вейвлет-преобразования в проблеме распознавания речи // Вестник ТГУ. – 2010. – Т. 5. – Вып. 6. – С. 1893–1899.
4. Smaragdis P., Raj B., Shashanka M. A Probabilistic Latent Variable Model for Acoustic Modeling [Электронный ресурс]. – Режим доступа: <http://www.merl.com/publications/docs/TR2006-121.pdf>, своб.
5. Личман Е.Ю. Элементарная теория музыки и основы творческого музицирования. Учебно-методическое пособие // Павлодар: ПГПИ. – 2010. – С. 48.
6. Зубаков А.П. Фурье и вейвлет-преобразования в проблеме распознавания речи // Вестник ТГУ. – 2010. – Т. 15. – С. 1893–1899.

7. Lacoste A., Eck D. Onset detection with artificial neural networks for mirex 2005 [Электронный ресурс]. – Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.119.3769&rep=rep1&type=pdf>, своб.
8. Алиев Р.М. Оптимизация оконного фурье и непрерывного вейвлет преобразований для спектрального анализа музыкальных сигналов // Цифровая обработка сигналов. – 2013. – № 2. – С. 16–19.
9. Cogliati A., Duan Z. Piano music transcription modeling note temporal evolution // Acoustics, Speech and Signal Processing (ICASSP). – 2015. – P. 429–433.
10. Шевченко Н.А. Распознавание последовательности аккордов в цифровом звуке: курсовая работа. – СПб.: СПбГУ, 2016. – 12 с.
11. Bello J.P. A Tutorial on Onset Detection in Music Signals // IEEE Transactions on Speech and Audio Processing. – 2005. – V. 13. – № 5. – P. 1035–1047.
12. Bello J.P., Daudet L., Abdallah S., Duxbury C., Davies M., Sandler M.B. A Tutorial on Onset Detection in Music Signals // IEEE Transactions on Speech and Audio Processing. – 2005. – V. 13(5). – P. 1035–1047.
13. Рябенко Е.А. Выбор функций потерь в задачах неотрицательного матричного разложения [Электронный ресурс]. – Режим доступа: http://www.machinelearning.ru/wiki/images/f/f6/AdaptiveNMF_CCAS.pdf, своб.
14. Дьяконов В.П. Фурье- и вейвлет-преобразования в компьютерной математике [Электронный ресурс]. – Режим доступа: <https://docplayer.ru/26693561-Fure-i-veyvlet-preobrazovaniya-v-kompyuterno-matematike-dyakonov-v-p.html>, своб.
15. Рябенко Е.А. Мультипликативный метод неотрицательного матричного разложения с АБ-дивергенцией и его сходимость // Машинное обучение и анализ данных. – 2014. – Т. 1. – № 7. – С. 800–816.
16. Dixon S. Onset detection revisited // Proc. of the 9th Int. Conference on Digital Audio Effects (DAFx-06). – 2006. – P. 133–137.
17. [Электронный ресурс]. – Режим доступа: http://mcstore.ru/chto_takoe_muzikalnij_zvuk.htm (дата обращения: 03.05.2017).
18. [Электронный ресурс]. – Режим доступа: <http://www.badlogicgames.com/wordpress/?p=187> (дата обращения: 03.05.2017).
19. [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/193514/> (дата обращения: 06.05.2017).

**Ярош Станислав Вадимович**

Год рождения: 1993

Университет ИТМО, факультет методов и техники управления
«Академия ЛИМТУ», кафедра компьютерного проектирования
и дизайна, студент группы № S4105Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: stanislav.yarosh@gmail.com

УДК 004.51**ИССЛЕДОВАНИЕ ПРОБЛЕМ СОЗДАНИЯ АНИМИРОВАННЫХ ИНТЕРФЕЙСОВ****Ярош С.В.****Научный руководитель – д.т.н., профессор Погорелов В.И.**

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

В работе получено представление о современном состоянии интерфейсной анимации. Выявлены проблемы, возникающие при создании анимированных интерфейсов. Выявлены критерии для дальнейшего проведения сравнительного анализа программных средств для создания анимированных интерфейсов.

Ключевые слова: анимация, пользовательский интерфейс, пользовательское взаимодействие, прототипирование, программные средства, дизайн.

В настоящее время программные продукты, разработанные для создания анимации, являются неотъемлемой частью рабочего процесса дизайнера, связанного с проектированием интерфейсов. Данные инструменты необходимы для того, чтобы быстро сделать прототип какой-то части интерфейса, а затем наглядно показать разработчику или заказчику то, как будет происходить взаимодействие с продуктом, не затрачивая при этом больших усилий (времени и денег) на разработку работающего приложения, системы или сайта.

С каждым годом увеличивается количество программных средств, которые стремятся соответствовать текущим тенденциям развития технологий и решить проблемы, возникающие при создании анимированных интерфейсов [1]. Зачастую вопрос выбора инструмента, подходящего для решения той или иной задачи, становится трудным, в особенности, для начинающих специалистов.

Анимация является отличным инструментом для создания «живых» интерфейсов, которые во всех смыслах лучше статичных картинок, которые были в прошлом. Сегодня интерфейс без анимации кажется пользователю недружелюбным, несовременным и даже «сломанным». Такой интерфейс уступает конкурентам, которые закладывают анимацию в дизайн своего приложения, сайта или сервиса [2].

Анимация в интерфейсе необходима для того, чтобы пользователь смог сориентироваться, осмыслить, что происходит на экране, заметить что-то важное, догадаться о чем-то, узнать что-то.

Существует четыре главные проблемы, возникающие при создании интерфейсной анимации, а также предлагаются методы их решения [3].

Первая проблема – технические ограничения. Дизайнерам приходится проектировать интерфейсы под различные размеры экранов и разные платформы (iOS/Android/Windows), они так или иначе сталкиваются с ограничениями вышеупомянутых платформ. Сложная анимация, которая может быть реализована для мобильных платформ без особых затрат

ресурсов, в вебе наоборот будет расходовать слишком много энергоресурсов, поэтому для начала, перед созданием анимации, следует:

- решить для какого типа устройств будет создаваться анимация (веб или Android/iOS/Windows);
- стоит ли использовать ресурсоемкую анимацию.

Вторая проблема – большой размер ресурсов, медленное интернет-соединение. Даже несмотря на оптимизацию графических ресурсов, задействованных в анимации, при «плохом» Интернете, анимация может не сработать или выглядеть «сломанной». Для решения этой проблемы предлагается:

- загружать «тяжелые» ресурсы в последнюю очередь;
- определять «медленный» Интернет и загружать облегченную версию анимации, либо вообще ее не использовать.

Третья проблема – анимация по-разному выглядит на разных устройствах. В настоящее время существует множество мобильных операционных систем, и в каждой операционной системе анимации элементов интерфейса происходят по-разному. Чтобы анимация выглядела согласованно на различных типах устройств следует:

- выбрать наиболее подходящую библиотеку для «родной» анимации;
- для мобильного веба использовать более быстрый Canvas.

Четвертая проблема – большой расход аккумулятора. Отрисовка анимации требует больших вычислительных ресурсов графического процессора. Чем больше анимации происходит на экране и чем она сложнее, тем больше затрачивается энергии, и ресурс аккумулятора резко снижается. Чтобы снизить большой расход энергоресурсов устройства предлагается:

- сменить реализацию или библиотеку анимации;
- снизить длительность анимации;
- если затрагивает ряд устройств, то отключить анимацию только для них;
- отказаться от анимации.

Можно выделить следующие критерии [4, 5], по которым можно будет определить подходит ли данный инструмент для решения той или иной задачи или нет:

- цена. Платный или бесплатный. Если платный, то по подписке, или покупается лицензия (единовременная оплата). Есть ли скидки для студентов, преподавателей университета или некоммерческих организаций;
- поддержка операционной системы. OS X, Windows или обе;
- режим работы. Онлайн (работа в браузере) или офлайн (работа на компьютере). В некоторых случаях может быть комбинированный вариант;
- форматы экспорта. Это может быть интерактивный прототип в браузере, который хранится в облаке, и каждый имеет доступ к нему по ссылке. Также могут быть форматы: GIF; любой видеоформат; HTML, CSS и JavaScript;
- порог входа. Прост в изучении или требует много времени и усилий для освоения;
- возможность экспорта анимации в нативный код платформы (iOS/Mac/Android);
- возможность создания шаблонов. Шаблоны позволяют дизайнерам ускорить время создания прототипа, используя уже заранее подготовленные компоненты взаимодействия из библиотеки;
- детальная настройка анимации. Присутствует редактор кривой временной функции, который позволяет точно настроить анимирование элементов интерфейса.

Анимация играет важную роль в дизайне пользовательских интерфейсов. Существует ряд проблем, возникающих при создании анимированных интерфейсов, предлагаются методы их решения. Выделены критерии для оценки программного средства для создания анимированных интерфейсов. В дальнейшем исследовании планируется провести более подробный анализ инструментов для создания интерфейсной анимации.

Литература

1. D'Silva P. The State of Interaction Design Tools [Электронный ресурс]. – Режим доступа: <https://medium.com/@pasql/the-state-of-interaction-design-tools-f755c6515368>, своб.
2. Head V. Designing Interface Animation. – Rosenfeld Media, 2016. – 240 p.
3. Фёдоров В. Школа мобильного дизайна: Анимация в интерфейсах [Электронный ресурс]. – Режим доступа: <https://www.youtube.com/watch?v=DTLF-Mpv534>, своб.
4. Head V. 4 tools for designing better UI animation and when to use them [Электронный ресурс]. – Режим доступа: [http:// motionographer.com/2016/10/31/4-tools-for-designing-better-ui-animation-and-when-to-use-them/](http://motionographer.com/2016/10/31/4-tools-for-designing-better-ui-animation-and-when-to-use-them/), своб.
5. Palmor T. All the UX prototyping tools Ever [Электронный ресурс]. – Режим доступа: <https://uxtools.co/blog/all-the-ux-prototyping-tools-ever/>, своб.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
Быкова А.А. Информационные и коммуникационные технологии в спорте.....	4
Волков В.О. Локальная вычислительная сеть мобильного узла связи на базе автомобиля КАМАЗ.....	8
Глебов Р.Г. Исследование методов и средств обеспечения безопасности в корпоративных сетях на примере мобильных устройств с использованием мобильного комплекса ViPNet	12
Кожин К.Р. Анализ систем управления контентом для разработки и продвижения web-сайтов	16
Кудрявых А.Ю. Особенности применения средств сетевого анализа и защита от них.....	20
Малярчук Л.К. Разработка информационной системы для учета, анализа и планирования вакцинации населения	24
Новикова Е.И. Исследование методов повышения производительности и масштабируемости прикладных решений 1С	28
Петров В.В. Использование программного обеспечения TICK, ELK, Zabbix для мониторинга информационных систем	31
Платунова С.М. Технологии объединения параллельных каналов передачи в сетях Ethernet	35
Тазетдинов А.А. Обоснование выбора базы данных для динамически изменяющихся объемов информации	37
Трубицин А.М. Исследование процесса безопасного использования сетевых протоколов	41
Алексейцева Е.В. Анализ зарубежных и отечественных исследований проблемы свободно распространяемого программного обеспечения для цифровой живописи	45
Ананьева А.А. Программные средства визуализации упаковки товара	48
Бабарицкий П.А., Сокуренок Ю.А., Государев И.Б. Анализ и имплементация методов парсинга веб-ресурсов. Анализ зарубежных и отечественных источников по теме исследования	51
Бабахин А.В. Сравнительный анализ подходов к разработке веб-интерфейсов.....	54
Баев А.А., Государев И.Б. Обзор исследований по проблеме проектирования многопользовательских игр на веб-платформе.....	57
Базаева К.А. Информационные технологии в музыкальном образовании.....	61
Бачурина Л.Р. Анализ структуры динамических визуальных систем.....	63
Башкирцева Е.И. Обзор рынка приложений и устройств в сфере mHealth.....	67
Бобрик О.А., Шалобаев Е.В., Шуклин Д.А. Обзор сервисов для визуализации данных ...	71
Букирёва Л.А. Исследование характеристик CMS на платформе Node.js	75
Ванина Н.В. Возможности нейронных сетей в распознавании цвета изображений.....	79
Варганов Д.А. Обзор постпроцессоров CSS и их задач.....	82
Водова В.А., Готская И.Б. Исследование технологий персонализации контента и пользовательского опыта в мобильных приложениях	84
Волобуев Н.Г. Сравнительный анализ представлений на примере платформ ASP.NET и Django	88
Волошин О.В. Разработка утилиты структуризации эвристических методов в производственных процессах	92
Воробьев А.И., Государев И.Б. Анализ проблем миграции баз данных веб-приложений от реляционной модели.....	96
Гаврилов Д.В. Исследование технологии печати символики киберспортивных организаций.....	98

Гарифуллин А.А., Сокуренок Ю.А. Моделирование процессов добычи полезных ископаемых на веб-платформе	101
Гарматина И.А. Влияние анимации интерфейсов на UX веб-сайтов	104
Головатая Е.Н. Исследование особенностей UI/UX дизайна мобильных приложений	108
Горбунов А.В. Сравнительный анализ сервисов облачного рендеринга	111
Давлетшин В.Р. К вопросу о проблемах и ошибках, существующих в текущих JavaScript-фреймворках	114
Деркунская С.А., Шуклин Д.А. Формообразующие факторы динамической айдентики – типографика и визуальный образ	117
Дмитриевская А.А. Веб-картография: состояние и перспективы развития.....	120
Довженко М.И. Анализ способов реализации алгоритма отслеживания изменений в одностраничных веб-приложениях	123
Дорофеева М.И., Погорелов В.И. Методы оценки юзабилити веб-интерфейсов.....	127
Дятлов Е.А. Модели анализа поисковой оптимизации веб-ресурса	131
Захарова А.О. Сравнительный анализ средств визуализации данных.....	135
Здоренко А.А. Проектирование и развитие фреймворков на платформе Node.js	139
Зиннатулин Ф.Ф., Шалобаев Е.В. Анализ исследований трехмерного моделирования в медицине.....	141
Ибрагимов Д.И. Обзор методологии представления контента различного типа в браузерной среде.....	144
Иванова П.В. Анализ исследований интерактивных книг для детей.....	147
Казначеева Е.О., Государев И.Б. Модульный и компонентный подходы в программировании.....	150
Калинина К.В., Шуклин Д.А. Подходы к проектированию веб-сервиса по аренде недвижимости.....	153
Кардашов С.А., Сокуренок Ю.А., Государев И.Б. Сравнительный анализ методологий прототипирования веб-ресурсов.....	156
Катафеев Д.Ю. Особенности использования методов геймификации в системах здравоохранения.....	159
Кемалжанов Д.И., Романова А. Технологии удаленного доступа к корпоративным информационным системам.....	162
Клочков С.В., Государев И.Б. Сравнительный анализ методологий веб-разработки на основе различных технологических стеков.....	164
Ковалев В.В. Обзор моделей предоставления облачных вычислений в вебе.....	167
Ковеченков В.О. Исследование имплементации методов машинного обучения в веб	171
Козак О.О., Зимина Д.В. Методы и средства моделирования технологий дистанционного обучения.....	173
Костюченко Ю.А. Анализ подходов к моделированию данных с помощью библиотек языка Python.....	175
Кузнецова Н.В. Обзор методик разработки на PHP	179
Линькова Т.И., Шуклин Д.А. Методики типографики в дизайне пользовательских интерфейсов веб-сайтов	182
Лисакова А.В. Сравнительный анализ возможностей фреймворков Twitter Bootstrap и Zurb Foundation	186
Литвинов К.В., Сокуренок Ю.А., Государев И.Б. Обзор теоретических и эмпирических основ веб-разработки для интернета вещей.....	190
Лобач И.Н., Шуклин Д.А. Использование анимационных эффектов в дизайне пользовательских интерфейсов веб-сайтов	194
Ломтев И.А. Исследование протоколов передачи данных для систем мониторинга	197
Лунев А.В. Алгоритмический дизайн, его особенности и недостатки, инструменты реализации	201

Магнитова М.В., Шуклин Д.А. Анализ и классификация критериев оценки коммерческих веб-ресурсов	204
Маревская М.В., Государев И.Б. Особенности использования веб-фреймворков при разработке десктопных приложений	208
Мартюшов М.В., Шуклин Д.А. Перспективные направления развития вспомогательных сервисов для продвижения веб-сайтов	211
Матушевский Н.Е. Определение оптимального алгоритма нечеткого поиска для нахождения дубликатов карточек пациентов в медицинской информационной системе	214
Миннемуллин Б.М., Погорелов В.И. Игрофикация в дистанционном обучении школьников.....	218
Молодцова В.С., Шуклин Д.А. Технология дополненной реальности в архитектуре	222
Мустафина Л.И. Использование библиотек для веб-анимации	225
Неудачин В.Е. Анализ источников по теме: «Исследование и сравнительный анализ методов генерации электронных валют»	227
Никитинская В.М., Сокуренок Ю.А., Государев И.Б. Обзор современных практик создания статических веб-сайтов	229
Нифанина А.Р. Особенности использования графики для продвижения товаров и услуг	232
Павлихин С.В. Сравнительный анализ механизмов обмена данными между веб-клиентом и веб-сервером	235
Парфенова О.В. Анализ популярности CMS-систем в 2017 году	239
Петрова Е.Ю., Государев И.Б. Анализ источников по теме: «Проблема расширения функциональности корпоративных порталов»	243
Полетова А.И. Анализ современных подходов к UX-дизайну мобильных приложений.....	246
Полякова Д.А. Анализ зарубежных и отечественных исследований проблемы адаптации 3D-моделей для 3D-печати	249
Романова А. Перспективы развития устройств виртуальной реальности	253
Ростов Д.С., Сокуренок Ю.А., Государев И.Б. Методы выбора программных средств разработки в зависимости от веб-интерфейса.....	255
Савенкова И.А. Особенности разработки геймифицированного приложения для операционной системы Android.....	258
Севостьянова Ю.С., Шуклин Д.А. Анализ научных исследований проблемы проектирования веб-приложения для взаимодействия поставщиков и потребителей	262
Сенин А.Д., Государев И.Б. Анализ зарубежных и отечественных источников по теме «Модели реактивного программирования в вебе и их имплементация»	266
Сенчило М.С. Исследование технологий алгоритмического дизайна Интернет-проектов.....	270
Сергейчук Н.А. Системы веб-аналитики: сферы применения и дальнейшие перспективы.....	273
Сивинский С.А. К вопросу расширения возможностей систем дистанционного обучения Moodle для формирования результата тестирования по различным критериям.....	276
Слепухин Я.И. Средства контроля и улучшения качества программного кода на языке JavaScript	280
Смирнов Д.Ю. Анализ использования серверного «рендеринга» при построении современных веб-приложений.....	283
Сохина А.А. Эволюция методов CSS-разметки веб-страниц	286
Ставрова В.С. Технологии оптимизации текстур для игровой трехмерной графики	290
Тодоровская Я.В., Шуклин Д.А. Обзор подходов к проектированию пользовательских интерфейсов	293

Туркунов А.А. Оценка практического внедрения технологий VR и AR для диспетчеризации и отслеживания систем/оборудования в зданиях общего пользования	296
Фирсова М.А., Сокуренок Ю.А., Государев И.Б. Обзор паттернов менеджмента зависимостей в JavaScript-фреймворках	299
Фомин А.А. Обзор методик динамической визуализации в веб-приложениях	303
Хренкова М.А. Теоретический обзор OpenSource-технологий проектирования элементов фирменного стиля	306
Чебыкина Д.А. Особенности автоматизации процесса верстки печатного издания	310
Челокьян Л.А. Подготовка модели изделия для последующей 3D-печати с учетом методов инженерного анализа	312
Шайхетдинова А.Р. Исследование методов поисковой оптимизации мобильных версий сайта	315
Шакирзянова Е.И. Социальные сервисы как инструмент SMM-продвижения образовательных услуг	317
Швалева Д.С., Сокуренок Ю.А., Государев И.Б. Анализ зарубежных и отечественных источников по теме «Сравнительный анализ методологий управления объектной моделью веб-документа»	320
Щербаков А.Б. Сравнительный анализ серверных фреймворков на платформе Node.js	323
Яковлева М.А. Обзор методов автоматической музыкальной транскрипции	326
Ярош С.В. Исследование проблем создания анимированных интерфейсов	330

**АЛЬМАНАХ НАУЧНЫХ РАБОТ
МОЛОДЫХ УЧЕНЫХ УНИВЕРСИТЕТА ИТМО
Том 7**

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 19.11.2018

Заказ № 4147

Тираж 100 экз.