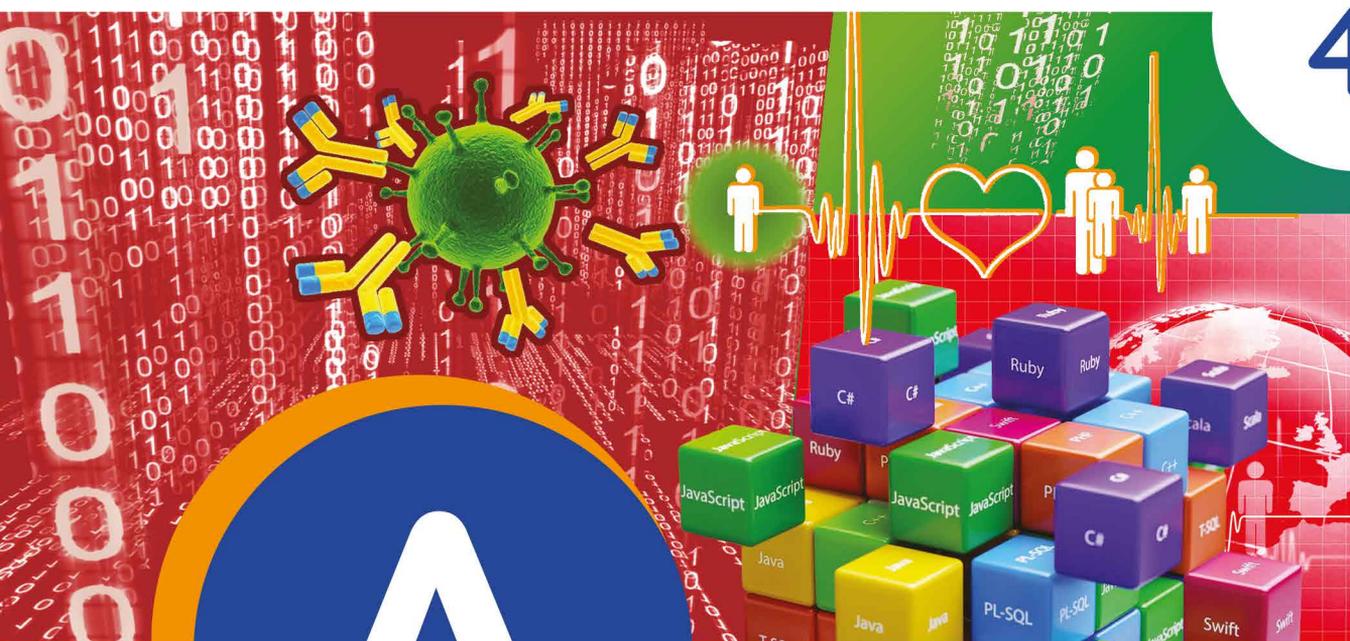




УНИВЕРСИТЕТ ИТМО

4



А

ЛЬМАНАХ

НАУЧНЫХ РАБОТ
МОЛОДЫХ
УЧЕНЫХ

2017

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

**АЛЬМАНАХ
НАУЧНЫХ РАБОТ
МОЛОДЫХ УЧЕНЫХ
Университета ИТМО**

Том 4



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург

2017

Альманах научных работ молодых ученых Университета ИТМО. Том 4.
– СПб.: Университет ИТМО, 2017. – 288 с.

Издание содержит результаты научных работ молодых ученых, доложенные на XLVI научной и учебно-методической конференции Университета ИТМО по тематике интернет-технологии и программирование.

ISBN 978-5-7577-0566-8

ISBN 978-5-7577-0570-5 (т. 4)



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.

© Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 2017

© Авторы, 2017

ВВЕДЕНИЕ

Издание содержит результаты научных работ молодых ученых, доложенные 31 января – 3 февраля 2017 года на XLVI научной и учебно-методической конференции Университета ИТМО по тематике интернет-технологии и программирование.

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.



Ампилогов Николай Сергеевич

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: kolyanwork@gmail.com

УДК 004.62

ТИПОВАЯ СТРУКТУРА API СОЦИАЛЬНЫХ СЕТЕЙ

Ампилогов Н.С.

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

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе рассматривалась структура API социальных сетей, на примере vk.com и facebook.com. Описаны основные правила запросов и получаемых ответов от сервисов.

Ключевые слова: api, социальные сети, vk, fb.

С каждым днем становится все больше приложений, которые построены на основе API какого-либо сервиса.

API (application programming interface) – это посредник между разработчиком приложений и какой-либо средой, с которой это приложение должно взаимодействовать. API упрощает создание кода, поскольку предоставляет набор готовых классов, функций или структур для работы с имеющимися данными [1]. Новые современные сервисы сначала пишут API, а уже потом создают клиенты под разные платформы.

У любого крупного, популярного сервиса в интернете есть свой API. Благодаря методам, предоставляемым интерфейсами, пишутся интересные и полезные приложения, автоматизируются рутинные и объемные процессы, причем не важно, под какую платформу написан клиент, скрипт, будь то web-клиент, настольное или мобильное приложение – все они будут выполнять и расширять базовую функциональность главного сервиса.

В работе пойдет речь о структуре API социальных сетей.

Социальные сети дают доступ к большому объему данных о пользователях и не только, поэтому наиболее интересны для анализа.

API ВКонтакте. API ВКонтакте – это интерфейс, который с помощью HTTP-запросов позволяет получать информацию из базы данных vk.com. На стороне сервиса уже определены синтаксис и тип запросов, ответов.

Синтаксис запроса. Чтобы обратиться к методу API ВКонтакте, достаточно выполнить POST- или GET-запрос вида:

`https://api.vk.com/method/NAME?PARAM&access_token= TOKEN&v=V`

Запрос состоит из нескольких частей:

- NAME – название метода API vk, к которому необходимо обратиться. Полный перечень методов доступен по ссылке <https://vk.com/dev/methods>;
- PARAM – параметры, входящие в состав метода API, ключ1=значение1&ключ2=значение2, данные параметры описаны на странице соответствующего метода;
- TOKEN – токен, уникальный ключ доступа. Необходим для большинства методов API. Подробное описание получения ключа <https://vk.com/dev/authentication>;

- V – используемая версия API, передается одновременно со всеми запросами. Актуальная версия – 5.62, по умолчанию используется – 3.0, для сохранения совместимости (в зависимости от версии, ответ от сервера может отличаться).

Параметры следует передавать:

- GET – если объем данных до 2 килобайт;
- POST – если объем данных более 2 килобайт.

Формат ответа по умолчанию – JSON, но также можно получать ответы в формате XML, для этого нужно добавить к названию метода «.xml» (пример: users.get.xml).

У API ВКонтакте есть определенные ограничения, все они указаны в документации на странице <https://vk.com/dev/manuals>.

API Graph (API Facebook). API Graph – это основной инструмент для загрузки и получения данных с платформы Facebook. Он представляет собой низкоуровневый API на основе HTTP-запросов, с помощью которого можно запрашивать данные, создавать публикации, управлять рекламой, загружать фото и выполнять множество других задач, необходимых для приложений [2]. Актуальная версия API Graph – v2.8.

Синтаксис

запроса.

https://graph.facebook.com/OBJECT?fields=FIELDS&access_token=TOKEN

Система представления данных на Facebook, состоит из следующих элементов:

- узлы – объекты (пользователь, страница, комментарий, фото...). Каждому узлу назначен уникальный ID;
- границы – соединение объектов (комментарий к фото, фото на странице...);
- поля – данные об объектах (id, имя, день рождения).

Для получения данных из API необходимо отправлять HTTP-запросы (GET) на узлы или границы между этими узлами.

Запросы нужно посылать на адрес graph.facebook.com, если нужно работать с видео, то используется адрес graph-video.facebook.com. Ответ приходит в формате JSON.

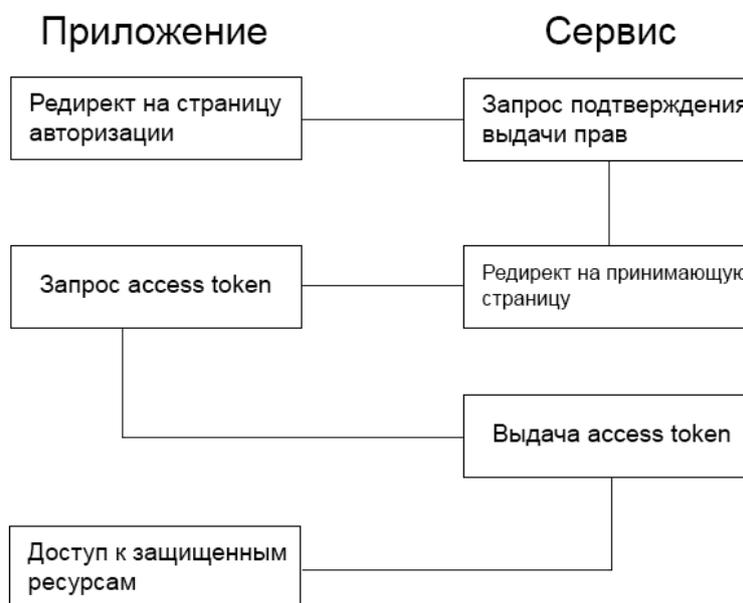


Рисунок. Взаимодействие приложения и сервиса

На основе вышеперечисленных данных, можно сделать выводы, что структура API социальных сетей схожа, но имеет свои особенности и ограничения. Для полноценной работы с API, необходимо получить уникальный ключ – access_token (рисунок) и через HTTP-запросы передавать необходимые методы, объекты, поля. Использование API сильно упрощает и ускоряет разработку качественных и полезных приложений и скриптов.

Литература

1. ВКонтakte / Разработчикам [Электронный ресурс]. – Режим доступа: <https://vk.com/dev/>, своб.
2. Документация Facebook / API Graph [Электронный ресурс]. – Режим доступа: <https://developers.facebook.com/docs/graph-api/>, своб.



Ананьева Александра Алексеевна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

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



Шуклин Дмитрий Анатольевич

Год рождения: 1979

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, к.педагог.н.

e-mail: do@limtu.ru

УДК 004.92

МЕТОДЫ ТЕСТИРОВАНИЯ ДИЗАЙНА УПАКОВКИ

Ананьева А.А., Шуклин Д.А.

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

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

В работе рассмотрены и проанализированы основные методы тестирования дизайна упаковки товаров. Выявлены их преимущества и недостатки.

Ключевые слова: компьютерная графика, маркетинг, тестирование дизайна, упаковка товара.

Ситуации, в которых рекомендуется проводить маркетинговое тестирование дизайна упаковки, определяются исходя из задач бренд-менеджмента. Наиболее распространены три случая:

- аудит существующей упаковки;
- создание новой упаковки;
- обновление существующей упаковки.

Для тестирования упаковки подходят не все методы маркетинговых исследований. Применение некоторых из них требует значительных временных и денежных затрат, а также может быть сопряжено со специфическими трудностями. В зависимости от выбранной методики заметно различается уровень адекватности и релевантности получаемых данных.

Все существующие исследовательские технологии подразделяются на две группы – качественные и количественные.

К первым относятся фокус-группы и глубинные интервью. Для тестирования дизайна упаковки они не очень подходят: выборка слишком мала и нерепрезентативна, а мнения и

вкусы отдельных людей слишком сильно влияют на конечный результат. При этом очень не многие респонденты способны объяснить, что действительно им нравится или не нравится в упаковке, или почему они предпочтут ту, а не другую.

Применение количественных исследований позволяет получить данные, отражающие реальное восприятие дизайна целевой аудиторией, для последующего анализа. На практике наибольшей эффективностью обладают два метода количественных исследований: личное интервью или опрос, а также *hall-test*.

Личное интервью – метод количественного исследования путем сбора информации в процессе непосредственного общения интервьюера с респондентами по структурированной анкете. Пригоден для тестирования упаковки, если необходим неглубокий анализ при небольшом количестве концептов, т.е. когда отсутствует возможность использовать дополнительное оборудование и проводить опрос по большой анкете.

Холл-тест (*in hall-test*) – классический метод маркетинговых исследований. Позволяет проводить формализованное личное интервью с элементами маркетингового эксперимента и применением специальных методик, использовать максимальное количество исследовательских инструментов без каких-либо ограничений по форме и сценарию интервью. Особенность холл-теста – возможность непосредственного контакта респондента с тестируемым объектом и проведения экспериментов с использованием специального оборудования: проектора для показа слайдов, макета торговой полки и т.д. Метод позволяет оценить концепт упаковки изолированно, а также сравнить концепты упаковки между собой и с конкурентами. Результатом холл-теста может быть комплексная оценка маркетинговой эффективности дизайна упаковки, включая визуальную силу, силу имиджа, эмоциональный отклик и предпочтения целевой аудитории [1].

Информативность количественных исследований, в частности, холл-тестов, можно существенно повысить, применяя специальные методики. Они позволяют получить наиболее полную и правдивую информацию по маркетинговой эффективности дизайна упаковки в удобной для анализа форме. Каждая из них направлена на изучение отдельной группы параметров, и каждое исследование может быть индивидуально сформировано из специальных блоков-методик. Существует ряд «классических» методик по тестированию дизайна упаковки в рамках холл-теста. В первую очередь к ним относятся:

- «Моделирование полки». Респонденту демонстрируют макет полки магазина и задают вопросы: «Какую упаковку вы заметили первой?», «Какая из них вам больше нравится?», «Какой из представленных образцов товара вы бы купили?»;
- методы определения оптимальной цены (*Price Sensitivity Meter*, *Brand Price Trade Off* и другие) – измерение чувствительности респондентов к цене и установление оптимальной цены на товар;
- игровые и проективные методики.

По данным компании *Package Testing Lab* из инновационных специальных методик наиболее действенны следующие:

1. *Visual Force*. Метрики визуальной силы дизайна показывают, насколько он способен привлечь к себе внимание и быстро быть опознанным в условиях реальной полки, доминировать в конкурентном окружении. *Visual Force* базируется на ряде метрик: трекинг взгляда, визуальная сила, тахистоскопия и другие [1, 2].

Наиболее надежный метод, достаточно полно и правдиво отражающий реальное восприятие потребителей, это тахистоскопия (*T-scope tests*). Методика адаптирована для тестирования упаковки из экспериментальной психологии. Тест предполагает использование проектора и специального программного обеспечения. Зрителю на короткий промежуток времени (от 0,007 с) демонстрируется концепт дизайна упаковки.

По реакции реципиента определяются:

- узнаваемость бренда;
- идентификация категории и продукта;

- узнаваемость логотипа и отдельных графических элементов;
- узнаваемость формы упаковки;
- ошибки, связанные с идентификацией.

Результаты исследования часто не нуждаются в дополнительной трактовке, свидетельствуя, почему тот или иной бренд продается лучше конкурирующих. Например, покупатели распознают категорию продукта и бренд чипсов Estrella на доли секунды быстрее, чем чипсы «Московский картофель». А позицию из вкусовой линейки Estrella, которую представляет упаковка, они распознают почти вдвое быстрее, чем у конкурента. Опционально данный тест можно провести для 2–3 упаковок основных конкурентов клиента.

2. Image Force. С помощью метрик Image Force (сила имиджа) можно оценить, что именно упаковка сообщает потребителю, какие эмоции и ощущения вызывают у него. Оценка показателя базируется на 20–30 ключевых атрибутах упаковки и методе семантического дифференциала. Под атрибутом понимается абстрактная или реальная характеристика бренда, которая может быть оценена респондентом. Ментальные атрибуты в данном списке должны соответствовать трем широким областям:

- эффективность продукта. Эти показатели атрибутов упаковки измеряют, до какой степени респондент верит, что непосредственно содержимое упаковки будет удовлетворять его ожиданиям;
- эстетический вид упаковки: параметры «привлекательная упаковка», «притягивающий внимание дизайн»;
- отражение личных интересов и ценностей респондента: «это мой тип продукта», «я буду использовать этот продукт каждый день».

Если показатель Visual Force отвечает за то, что упаковку заметят и правильно идентифицируют на полке, то показатель Image Force формирует силу продажи. В результате теста мы получаем свойства, совпадающие с ожиданиями потребителей, а также свойства, дифференцирующие одну упаковку от другой. Это, например, воспринимаемое качество продукта, воспринимаемая эффективность продукта, эстетика упаковки, соответствие ценностям потребителя.

3. Preference Force. Метрики силы предпочтения (Preference Force) показывают, по каким параметрам упаковка данного бренда выигрывает по сравнению с упаковками конкурентов. Метод основан на прямом сравнении упаковок и их преимуществ. Список утверждений для сравнения часто включает такие параметры как: наиболее экономичный продукт, наиболее эффективный продукт, наиболее привлекательная упаковка, продукт лучшего качества. Они должны выявить различия в том, как аудитория воспринимает упаковки различных брендов, а главное – как потребитель определяет «лучший выбор». Так, результаты тестирования упаковок шоколадных батончиков среди целевой аудитории, студентов московских вузов, выявили, что, например, Bounty – это «самый вкусный продукт». Наиболее ярко выраженные качества бренда Picnic – «лучше других утоляет голод» и «наиболее дешевый продукт из представленных». А Kit-Kat – «самый вкусный продукт», «наиболее качественный продукт» и «самый лучший продукт в целом». Для максимально эффективного воздействия упаковки на потребителя три показателя: Visual Force, Image Force и Preference Force должны быть сбалансированы. Каждый из них формирует свою часть цепочки потребительских реакций, от внимания до действия, т.е. покупки, в американском маркетинге именуемой АИДА [3, 4].

Мощный коммуникационный импульс бренда, транслируемый с магазинной полки и подкрепленный вспомогательными элементами дизайна, которые создают позитивное впечатление от наблюдаемого объекта, помогает достичь желаемого уровня продаж.

Упаковка – лицо большинства брендов. Покупатели судят по ней о качестве и цене продукта, дизайн упаковки транслирует позиционирование и ценности марки, эмоциональные выгоды и многие другие характеристики. Инвестиции в дизайн упаковки –

прямые инвестиции в имидж и продажи бренда. Компании могут тратить деньги на рекламу, ATL-мероприятия и мерчандайзинг, но без правильного дизайна упаковки эффективность подобных инвестиций сомнительна.

Литература

1. Моисеева Н.К., Рюмин М.Ю., Слушаенко М.В. Бренддинг в управлении маркетингом. – М.: Омега-Л, 2006. – 336 с.
2. Хайн Т. Все об упаковке. – М.: Арт-Родник, 2007. – 288 с.
3. Херриот Л. Библия упаковки. – РИП-Холдинг, 2007. – 304 с.
4. Энджел Д., Блэкуэлл Р., Минард П. Поведение потребителей. – СПб.: Питер, 2000. – 944 с.



Анацкая Елизавета Владимировна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: anatskayalisa@gmail.com



Государев Илья Борисович

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

e-mail: ilia.gossoudarev@gmail.com

УДК 004

ЭВОЛЮЦИЯ ВЕБ-АНИМАЦИИ

Анацкая Е.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 представлены этапы становления технологий реализации анимации в вебе. Рассмотрены наиболее популярные и современные подходы, применяемые в настоящее время. Приведен наглядный сравнительный листинг реализации анимации. Включены дополнительные источники, необходимые для более полного изучения темы, послужившие информационным базисом при проведении исследования.

Ключевые слова: анимация, интерактивность, динамика, GIF, JavaScript, CSS3, Web Animations API.

С появлением сети Интернет и возможности обмена информацией посредством веб-сайтов, возникает необходимость в грамотной, интуитивно-понятной и привлекающей пользователя форме представления данных. Сайты перестают быть лишь текстовой информацией, появляется необходимость в наделении страниц интерактивностью. Это

приводит к потребности в создании анимации, которая при правильном ее применении решает поставленную задачу.

Понятие анимации в вебе. Анимация – естественный шаг в эволюции веб-технологий и дизайна, представляющая цифровой мир в новой интуитивно-понятной форме. Результаты исследований демонстрируют анимацию как инструмент, позволяющий лучше понять происходящее на экране [1]. Анимация представляет собой визуальное отображение изменений скорости в пространстве и во времени и делится на три типа:

1. статическая анимация. Постоянное движение от начального состояния к конечному, без изменения промежуточных состояний;
2. анимация с учетом состояний. Возникает при воздействии на элемент, т.е. при изменении состояния анимируемого объекта;
3. динамическая анимация. Состоит из совокупности промежуточных состояний, являющихся результатом реакции на определенные события или действия пользователя.

Эволюция анимации. В настоящий момент интернет-ресурсы изобилуют множеством сложных эффектов, реализованных посредством встраивания анимаций на страницы сайтов и мобильных приложений. Однако так было не всегда. Анимация развивалась наряду с развитием веб-технологий.

GIF (Graphics Interchange Format). Первым инструментом для создания интерактивности на сайтах послужила GIF-анимация. Эффект движения обеспечивался быстрой сменой множества картинок. Однако в период возникновения данной технологии анимация еще не рассматривалась как средство улучшения удобства взаимодействия с сайтом, а скорее применялась для украшения или развлечения.

С появлением новых потребностей: использование анимации для улучшения навигации по сайту, создание лаконичного и удобного интерфейса взаимодействия с пользователем, стало понятно, что GIF-анимации исчерпала себя. Пришло время новых инструментов в мире веб-анимации.

Flash. Adobe Flash представляет собой мультимедийную платформу компании Adobe Systems, позволяющую создавать анимации, рекламные баннеры и игры, а также внедрять видео- и аудиозаписи на веб-страницы [2].

Популярность данной технологии объясняется ростом пользовательского интереса. Flash выводил простую анимацию на новый уровень, расширяя и дополняя ее. Оригинальность дизайна, которого можно было добиться посредством Flash и удобство работы с технологией привлекали внимание пользователя, обеспечивая интерес к посещению веб-ресурса [3]. Преимущества обеспечивали рост популярности технологии, а как следствие – рост продаж услуг продукции.

Однако наряду с совершенствующимися во много раз быстрее браузерными технологиями, Flash начал отставать. Среди недостатков:

- медленная загрузка страниц;
- неидентичное и некорректное отображение контента на всех устройствах;
- уязвимость.

Flash-технология предоставила универсальную среду для выполнения анимации любого типа и платформу для их редактирования, однако создана она была в эру персональных компьютеров и проявила свою несостоятельность, обличив явные минусы и неудобства после наступления эры отзывчивых интерфейсов и открытых веб-стандартов, мобильных устройств и гаджетов, для которых важным являлось сохранение заряда батареи [4]. Это была революционная технология для своего времени, однако созданные с помощью нее анимации были тяжелыми и в разы увеличивали время загрузки страниц.

JavaScript и CSS3. Современные подходы создания анимации. Возможности, предоставляемые браузерами, становятся все совершеннее. Это позволяет создавать и поддерживать анимацию без установки дополнительных плагинов, таких как Flash. В настоящее время для реализации интерактивного контента используется комбинация JavaScript, CSS3 и HTML5. Однако перед разработчиками встает вопрос о выборе подхода создания анимации: императивный (JavaScript) или декларативный (CSS).

1. JavaScript. Императивные анимации. Анимация, реализованная на JavaScript, – это код, выполняемый в одном потоке с другими скриптами. Поскольку поток может быть занят уже выполняющимся кодом, возможны конфликты, что зачастую служит причиной отсутствия кадров анимации. Преимущество анимации посредством JavaScript – это возможность контроля на каждом этапе. Анимацию можно замедлять, останавливать, запускать в обратном направлении и выполнять с ней другие манипуляции [5];
2. CSS. Декларативные анимации. Возможность оптимизации анимации самим браузером является важным преимуществом данного способа. Достигается это посредством вынесения некоторых операций на отдельные слои вне потока. Данного вида анимация применяется для реализации переходов и несложного интерактива. Однако минус состоит в том, что в отличие от JavaScript, CSS не обладает такой мощностью и динамичностью.

За последние несколько лет анимация получила значительное развитие благодаря поддержке CSS и JavaScript. Появились такие библиотеки как jQuery, Velocity.js и GSAP. Однако наряду с множеством преимуществ каждого из подходов, существуют минусы и нерешенные задачи оптимизации:

- производительность при выполнении анимации;
- обновление анимации и синхронизация;
- реализация аппаратного ускорения для переходов со встроенной поддержкой в браузере;
- реализация анимации без подключения сторонних библиотек.

Web Animation API. Тенденции развития анимации. На сегодняшний день инновационный прорыв в области анимации состоит в объединении декларативного и императивного подходов. Для этой цели разработан Web Animations API. Его назначение – совмещение производительности CSS с гибкостью JavaScript и SVG-анимации, а также реализация отслеживания качественного выполнения анимаций самим браузером.

Web Animation API – спецификация W3C. Синтаксис похож на CSS, но включает использование переменных, элементов управления и вызов функций. Реализованы следующие возможности:

- предоставление доступа к API движка анимации, что позволяет использовать инструменты браузера для создания и отладки анимации, а также повышать производительность при использовании библиотек;
- выполнение анимации в собственном потоке;
- регулирование траектории движения анимации;
- поддержка вложенных и последовательных анимаций;
- возможность различных манипуляций над анимацией с помощью опций управления временем и объекта AnimationPlayer.

На данный момент технология реализована еще не во всех браузерах, поэтому использовать ее нужно с осторожностью, заменяя альтернативными вариантами в неподдерживаемых браузерах. На текущий момент поддержка доступна в Google Chrome (49), Firefox (50), Opera (42), Android Browser (53), Chrome for Android (55). В скобках указаны версии, с которых начинается поддержка браузерами технологии Web Animations API.

Структура создания анимации посредством двух современных подходов, таких как Web Animations API и CSS, схожа, следовательно, переход с одной технологии на другую при необходимости использования требуемого функционала не представляет сложности.

Выводы. Исходя из вышеописанного, следует, что при необходимости внедрения интерактивности на сайт или в веб-приложение, руководствоваться выбором технологии реализации анимации необходимо исходя из поставленных задач и уровня сложности анимаций. Внимание следует уделить таким технологиям как CSS3 и Web Animation API. Таким образом, проведенное исследование с подробным рассмотрением различных технологий реализации анимации позволило определить наиболее рациональные и современные способы ее представления.

Литература

1. Состояние веб-анимации в 2014 году. [Электронный ресурс]. – Режим доступа: <http://frontender.info/the-state-of-animation-2014/#webanimationapidelvelichayshiydelapiokotoromvnikogdaneslshali>, своб.
2. Adobe Flash. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Adobe_Flash, своб.
3. Flash-анимация. [Электронный ресурс]. – Режим доступа: <http://qb-art.ru/flash-animatsiya.html>, своб.
4. Thoughts of flash. [Электронный ресурс]. – Режим доступа: <https://www.apple.com/hotnews/thoughts-on-flash/>, своб.
5. Web Animations API. [Электронный ресурс]. – Режим доступа: <https://htmlhook.ru/element-web-animations-element-animate.html>, своб.



Аринчихина Ирина Валериевна

Год рождения: 1973

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра управления и права, группа № S4215

Направление подготовки: 38.04.02 – Менеджмент

e-mail: Maruska-N@yandex.ru

УДК 34.08

ИССЛЕДОВАНИЕ КАДРОВОГО ПОТЕНЦИАЛА НА АВТОТРАНСПОРТНОМ ПРЕДПРИЯТИИ

Аринчихина И.В.

Научный руководитель – к.юрид.н., доцент Воронина М.В.

Работа является кратким обзором исследования кадрового потенциала на отдельно взятом автотранспортном предприятии.

Ключевые слова: экономика, кадры, потенциал, управление персоналом, сотрудники, предприятие.

Грузовой автомобильный транспорт России выполняет около 80% общего объема грузовых перевозок. Ему принадлежит большая роль в обеспечении перевозок для экономики страны [1–4].

Автомобильный транспорт не производит товаров, его продукцией является сам процесс перевозки, поэтому товаром транспорта являются комплексные транспортные услуги, предоставляемые клиентуре (перевозка грузов, погрузочно-разгрузочные работы, экспедирование, техническое обслуживание и ремонт подвижного состава и др.). Преимуществами автомобильного транспорта являются высокая маневренность, большая провозная способность, быстрота доставки грузов и пассажиров, меньшая себестоимость перевозок на короткие расстояния по сравнению с водным и железнодорожным транспортом и

некоторые другие. Большие скорости движения на усовершенствованных дорогах позволяют более быстро доставлять грузы и пассажиров, чем по водным и железнодорожным путям.

Доля автомобильного транспорта в перевозках непрерывно увеличивается.

В настоящее время Россия выходит на уровень экономической и финансовой стабилизации, но для быстрого развития предприятия этого мало. Необходим стремительный научно-технологический прогресс, который в первую очередь зависит от кадрового потенциала. В настоящее время наблюдается повышение спроса на специалистов в области управления персоналом, основная цель которых состоит в обеспечении успешной работы предприятия в условиях рыночной экономики.

Основной долгосрочной целью любого предприятия является стремление выжить в конкурентной борьбе. При этом сотрудники являются главным конкурентным преимуществом любого предприятия. Оценивая сочетание таких сложных характеристик, как максимальное использование возможностей, способностей и навыков сотрудников, достижение организационных целей и умение привлечь наиболее подготовленные и высокомотивированные кадры, можно судить об эффективности работы предприятия.

Все эти характеристики должны сказаться на общих показателях эффективности деятельности предприятия: производительности труда, качестве, уровне обслуживания клиентов, росте прибыли. Чем шире возможности и способности сотрудников, чем больше их навыки, тем стремительней предприятие достигает своих целей.

От человеческих ресурсов зависит слабость и мощь предприятия. Принимаемые сотрудниками на любом рабочем месте профессиональные решения обуславливают эффективность осуществления производственных задач и формируют общую успешность и результативность деятельности предприятия. Следовательно, важно контролировать, побуждать и помогать сотрудникам, чтобы не только создать все условия для наиболее полной реализации их профессионального потенциала, но и сформировать желание трудиться качественно и производительно. В том, как правильно управлять персоналом и добиваться повышения результативности деятельности предприятия, немалое место занимает оценка кадрового потенциала предприятия.

Важнейшим направлением кадровой политики сегодня становится повышение профессиональной квалификации персонала, что будет способствовать созданию профессионального, стабильного и авторитетного для общественного мнения коллектива.

Кадровая политика требует концептуального подхода, необходимо создать систему, позволяющую оптимально управлять кадровым потенциалом предприятия.

Планирование персонала предполагает определение потребности в кадрах. Оно должно исходить из оптимального состава рабочей силы и создания условий для успешной реализации личностного потенциала сотрудников, использования их навыков, опыта, квалификации.

При определении потребности в персонале необходимо исходить из принципа, согласно которому численность и качество персонала должны быть рассчитаны так, чтобы обеспечить долгосрочное выполнение задач предприятия. Его нехватка ставит под угрозу выполнение задач предприятия, а избыток вызывает излишние расходы, ухудшает позиции предприятия в конкурентной среде и тем самым ставит под угрозу его существование.

Потребность в кадрах, выявленная в результате планирования, определяет задачи управления персоналом – либо удовлетворение спроса (при недостатке), либо высвобождение (при избытке).

Основной задачей оценки кадрового потенциала является улучшение результатов работы, как отдельных исполнителей и отделов, так и предприятия в целом.

Литература

1. Богданова Е.А. Маркетинговая концепция организации персонал-менеджмента и конкурентоспособной рабочей силы. – СПб.: Изд-во СПбУЭФ, 1996.
2. Дафт Р.Л. Менеджмент. – СПб.: Питер, 2006. – 864 с.
3. Денисов А.Ф. Приключения кадрового менеджмента в России // Сборник тезисов 3-ей Всероссийской конференции. – 2000. – 20 с.
4. Спивак В.А. Управление персоналом: практикум по курсу. – СПб.: ИВЭСЭП, «Знание», 2008. – 144 с.



Афаунова Данна Рамазановна

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4206

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: 1d2a3n4a@mail.ru



Готская Ирина Борисовна

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, д.педагог.н.,
профессор

e-mail: iringot@mail.ru

УДК 004.921

СИМВОЛИЧЕСКАЯ СИСТЕМА ЕМОJИ КАК ИНСТРУМЕНТ ИЗУЧЕНИЯ ИНОСТРАННОГО ЯЗЫКА

Афаунова Д.Р.

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 этапы развития коммуникации, сравнительный анализ Еmojі с первыми наскальными рисунками, а также предложены методы применения Еmojі в сохранении языков малых народов России.

Ключевые слова: Еmojі, пиктограмма, вокабулярий, коммуникация, иконка, идеограмма, Unicode.

Современная система образования переживает ряд важных трансформаций, которые меняют процесс обучения детей миллениума. Для эффективного взаимодействия с поколением социальных сетей, онлайн-игр и виртуальной реальности необходимо говорить с ними на одном языке. Одновременно с этим в 2015 году Оксфордский словарь назвал иконку «лицо со слезами радости» символической системы Еmojі словом года, тем что, по сути, словом не является. Как это объяснить? Движемся ли мы к упрощению и деградации, или же это естественный процесс развития коммуникации человечества? Попробуем вспомнить и проанализировать, с чего все началось.

Около 17300 лет назад, когда не существовало современного языка общения, наши предки общались посредством наскальных рисунков. Они изображали на стенах пещер животных,

людей, звезды и многое другое. Люди обменивались внутренними инстинктами, а не осознанными мыслями. Человек видел определенный набор рисунков и принимал решения, основываясь на том, что он чувствует. Спустя несколько тысяч лет, рисунки приняли определенный порядок, который влиял на смысл восприятия информации. Человек больше не испытывал эмоцию смотря на каждый рисунок в отдельности. Необходимо было просмотреть весь набор картинок, прочитать их до конца строки, а затем анализировать прочитанное. Люди начали записывать, а не изображать мысли. Произошло разделение сознания на мысли и чувства. Затем, на протяжении следующих нескольких тысяч лет, коммуникация была упрощена от порывов творчества до пиктограмм, которые впоследствии стали основой алфавита или же идеограмм (иероглифов), символов, презентующих идею, появилась письменность.

С развитием письменности, науки и техники, разделение работы мозга на творческую и аналитическую часть значительно усугублялось. Однако с приходом с XX века, настал период регенерации. Например, художники футуристы в своих произведениях ярко показывали желание понимать и чувствовать одновременно. В комбинации языка и образа, они видели будущее и то, как мы будем мыслить и выражать свои чувства. Затем настал век рекламы, у которой было все необходимое (яркая картинка и короткий призывающий текст), чтобы оказывать сильное воздействие. За рекламой пришел век поп-культуры и комиксов. Возник новый способ обработки информации, понимания, чувствования. Наше сознание привыкло к такому способу восприятия, что подводит нас к цифровому экрану – новой наскальной доске. Очевидно, что выражать мысли и чувства в едином сообщении было естественно для человека задолго до Еmojі.

Символическая система Еmojі появилась в 1990-х годах в Японии.

В переводе с японского звук «е» – обозначает «картинка» и слово «тоjі» – «буква». Несмотря на то, что Еmojі имеет колоссальную популярность по всему миру, ее использование и смысл многих символов сильно искажены. Например, иконка «белый цветок» в японской культуре означает «блестяще выполненное домашнее задание», а изображения традиционных японских блюд Onigiri и Narutomaki используются на западе, как стейк и салат [1]. Исходя из этого, а также отсутствия какой-либо грамматики, очень рано говорить об универсальности этого языка. Система Еmojі не может дать пользователям свободу экспериментировать и создавать новые формы речи, так как любое ее преобразование напрямую зависит от обновления таблицы ASCII и международных компьютерных стандартов, регулируемых международным консорциумом Unicode [2]. Однако это не мешает использовать ее во вспомогательных целях коммуникации, знакомства с другой культурой или обучения.

Отличным примером стал эксперимент, проведенный в Швеции. Было создано мобильное приложение с большим набором Еmojі-иконок, выражающих чувства людей, подвергшихся физическому или иному насилию, для подростков и социальных работников. То, о чем подростки не могли рассказать напрямую, они могли показать с помощью Еmojі в приложении. За сутки в шведском AppStore приложение скачали более 10000 раз [3]. Успешный пример использования Еmojі в обучении (emojiscience.com), контент которого представлен в виде Еmojі-таблицы. Учитель естествознания Билл Най объясняет с их помощью научные концепции. Можно считать, что подобный способ обучения упрощает комплексные и сложные темы, как глобальное потепление, но знакомые из жизни иконки и значки, все же задерживают внимание детей сильнее, чем формулы, и на доступном языке объясняют сложные явления, понимание которых может надолго сохраниться в памяти.

Как можно использовать Еmojі в изучении и сохранении языков малых коренных народов России? Речь идет о молодом поколении людей, плохо владеющих родным языком в силу современного образа жизни. Чтение и письмо – это приобретенное качество каждого поколения, но образное восприятие информации заложено в нас с рождения. Классическую клавиатуру Еmojі можно использовать в изучении вокабулярия базовых слов, не имеющих двойственное значение. Почему изучение вокабулярия подобным способом может быть

полезнее обычных картинок или фотографий? Согласно исследованиям (University of British Columbia [4]) люди реагируют на плакаты с Еmojі так, как если бы на них были изображены реальные люди, ассоциируют себя с часто используемыми иконками. Также было доказано, что определенные Еmojі способны влиять на когнитивное мышление, что помогает проще запомнить сообщение. Второй шаг – создание собственной системы Еmojі, основанной на понятиях и словах того малого народа, чей язык изучается.

Современный феномен Еmojі может быть воспринят двояко: позитивно или же негативно, зависит от его применения. Глобализация, которая стирает идентичность и самобытность многих народов, может ровно столько же помочь сохранить разнообразие культур и языков в цифровом веке.

Литература

1. Samantha Lee. What communicating only in emoji taught me about language in the digital age? QUARTZ [Электронный ресурс]. – Режим доступа: www.qz.com/765945/emojis-forever-or-whatever-im-a-poet/, своб.
2. Joan Gage. Are Emoticons And Emojis Destroying Our Language? The Huffington Post [Электронный ресурс]. – Режим доступа: www.huffingtonpost.com/joan-gage/emoticons-and-emojis-destroying-our-language_b_7950460.html, своб.
3. Neil Cohn. Will emoji become a new language? BBC Future [Электронный ресурс]. – Режим доступа: www.bbc.com/future/story/20151012-will-emoji-become-a-new-language, своб.
4. Andrea Gonzalez. Emoji Pedagogy – Engaging Digital Learners with Emojis. The University of British Columbia (Vancouver Campus) [Электронный ресурс]. – Режим доступа: www.ets.educ.ubc.ca/emoji-pedagogy-engaging-and-educating-digital-learners-with-emojis/, своб.



Барташев Артем Николаевич

Год рождения: 1993

Факультет информационных технологий и программирования,
кафедра информационных систем, группа № М4205

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: ar.bartashev@gmail.com

УДК 004.021

СЕРВИС ДЛЯ РЕШЕНИЯ ЗАДАЧ ПЛАНИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ ДИАГРАММЫ ГАНТА НА ОСНОВЕ СИСТЕМЫ JIRA

Барташев А.Н.

Научный руководитель – к.т.н., доцент Хлопотов М.В.

Различные подходы и инструменты планирования проектов стали неотъемлемой частью успешного функционирования большинства компании. Одним из наиболее часто используемых подходов планирования проектов является диаграмма Ганта. В работе рассмотрен алгоритм построения диаграммы Ганта в рамках системы управления проектами, используемой в распределенных командах.

Ключевые слова: информационные системы, диаграмма Ганта, расчет расписания работ, алгоритмы на графах, системы управления проектами.

Существует огромное количество инструментов управления проектами, однако в последнее время все большую популярность набирает система Atlassian JIRA [1]. Данная система изначально позиционировалась, как система отслеживания задач, однако в

последние годы функциональность данной системы стала расширяться, во многом благодаря обширной базе плагинов, и ее стали применять тысячи компаний по всему миру для управления проектами.

Одним из таких плагинов, позволяющих использовать JIRA, как систему управления проектами, является плагин Structure [2]. Данный плагин позволяет отображать задачи из различных проектов в одном месте и представлять их в иерархическом виде. Однако на данный момент плагин не предоставляет возможности работать с диаграммой Ганта [3].

В настоящей работе рассмотрен алгоритм построения диаграммы Ганта, который необходим для создания системы, позволяющей работать с диаграммой Ганта на основе иерархии задач из плагина Structure.

В системе присутствуют два типа задач:

1. обычная задача – задача, не имеющая подзадач;
2. группа – задача, имеющая подзадачи (задачи-потомки).

Алгоритм построения диаграммы Ганта состоит из трех основных этапов:

Этап 1. Построение графа задач – граф – это наиболее удобный способ представления задач для расчета расписания. Узлы представляют задачи, а ребра – зависимости между задачами. По графу легко понять, какие задачи от каких зависят, и производить расчет в нужном порядке. Граф строится на основе списка задач, получаемого из Structure, и списка зависимостей между задачами из JIRA.

Этап 2. Поиск и удаление циклов в графе задач – расчет расписания возможен только по ациклическому графу, что связано с тем, что для расчета расписания задачи необходимо рассчитать расписание задач, от которых она зависит, а если в графе присутствуют циклы, то это выполнить невозможно.

Выполнение данной задачи осложняется двумя особенностями, а именно: в графе могут присутствовать два типа зависимостей – обычные зависимости и групповые зависимости, а также в графе может быть дублирование задач. Групповые зависимости – это зависимости между группой и подзадачами, обычные зависимости – это все остальные зависимости. Групповые зависимости образуют цикл, так как для расчета задачи, необходимо рассчитать расписание группы, а для расчета расписания группы необходимо рассчитать расписание всех потомков. Однако такие циклы можно обойти, модернизировав модель данных. Дублирование задач связано с тем, что система будет использовать список задач из Structure, одна и та же задача может встречаться в графе несколько раз и состоять в различных группах, однако расписание всех узлов, относящихся к одинаковой задаче, должно совпадать.

В связи с этими особенностями, обычные алгоритмы поиска циклов напрямую неприменимы. Для решения данной задачи был разработан алгоритм, основывающийся на алгоритме поиска в глубину [4]. Во-первых, для работы алгоритма была произведена модернизация изначальной структуры данных, а именно: список ребер каждого узла был расширен списком ребер узлов, ассоциированных с такими же задачами, что позволило решить проблему с циклами, возникающими в связи с дублированием задач в графе. Далее в каждый узел графа был добавлен новый тип ребер, который представляет групповые зависимости (между задачей и подзадачей), также группы были разбиты на два отдельных узла: начало группы и окончание группы, так как для расчета расписания потомков достаточно только начала группы, это позволило избавиться от циклов, связанных с групповыми зависимостями. После проделанных изменений структуры данных, стало возможным использовать модернизированную версию поиска в глубину, разбитую на два этапа: поиск циклов по групповым зависимостям и поиск циклов по обычным зависимостям.

Этап 3. Расчет расписания – расчет расписания представляет собой расчет начала и окончания выполнения всех задач на основе ациклического графа задач, полученного на предыдущем этапе, и параметров задач. К параметрам задачи относятся: тип задачи (группа или нет), время выполнения, рабочий календарь, часовой пояс и ограничения на задачи.

Рабочий календарь описывает набор временных промежутков, в рамках которых задача может выполняться. Пример рабочего календаря – рабочие дни с понедельника по пятницу, рабочее время с 10:00 до 19:00. Важной особенностью разрабатываемой системы является то, что задачи могут иметь различные календари и различные часовые пояса. Это связано с тем, что система JIRA может использоваться с распределенными командами, где люди, выполняющие задачи, могут находиться в различных городах и странах.

Расчет обычной задачи состоит из двух этапов: расчет начала выполнения и расчет окончания выполнения. Начало выполнения задачи рассчитывается за четыре шага:

1. рассчитывается расписание всех задач, от которых данная задача зависит;
2. время окончания выполнения всех задач и дата начала проекта переводится в часовой пояс данной задачи;
3. выбирается самое позднее время из величин, рассчитанных на предыдущем шаге, а также из ограничений на задачу;
4. временем начала выполнения задачи выбирается ближайшее рабочее время, позже времени, рассчитанного выше. Данный шаг необходим, так как у задач могут быть различные календари, и другая задача может закончиться в нерабочее время для данной задачи.

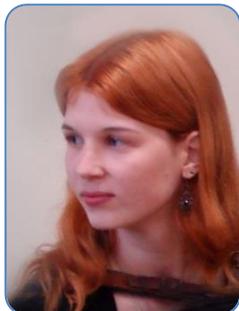
Окончание выполнения задачи рассчитывается на основе начала выполнения, рассчитанного ранее, на основе времени выполнения и рабочего календаря. К началу выполнения добавляются рабочие промежутки до того момента, как сумма рабочих промежутков станет равна времени выполнения задачи.

При расчете расписания группы, сначала рассчитывается начало выполнения группы как с обычной задачей, после этого рассчитываются все подзадачи группы и потом расписание группы пересчитывается так, что начало выполнения группы становится началом выполнения самой ранней подзадачи, а окончание выполнения – окончанием выполнения самой поздней подзадачи.

В работе приведен алгоритм для построения диаграммы Ганта в рамках системы с различными рабочими календарями и часовыми поясами, а также с рядом особенностей, таких как групповые зависимости и дублирование задач. Дальнейшие исследования направлены на расширение алгоритма, позволяющее учитывать доступные ресурсы для выполнения поставленных задач.

Литература

1. Atlassian JIRA [Электронный ресурс]. – Режим доступа: <https://www.atlassian.com/software/jira>, своб.
2. Structure – Enterprise Issue Organizer [Электронный ресурс]. – Режим доступа: <http://almworks.com/structure/overview.html>, своб.
3. White D., Fortune J. Current practice in project management – an empirical study // International Journal of Project Management. – 2002. – V. 20. – № 1. – P. 1–11.
4. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ. – 2-е изд. – Изд-во: Вильямс, 2012. – С. 470–477.



Бачурина Людмила Романовна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: lusya.bachurina@gmail.com

УДК 004.9

АНАЛИЗ ТЕХНОЛОГИЙ ГЕНЕРАТИВНОГО ДИЗАЙНА

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 сформулированы основные подходы к реализации проектов в области генеративного дизайна, рассмотрены возможности применения генеративных технологий и методов проектирования в графическом дизайне.

Ключевые слова: генеративный дизайн, параметрические методы проектирования, алгоритмический дизайн.

Генеративные технологии активно применяются в различных областях (аудиовизуальные искусства, трехмерная графика, архитектура, графический дизайн, промышленный дизайн, мода). Специфической особенностью генеративного дизайна является то, что результат проектирования (изображение, звук, архитектурные модели, анимации) порождается системой алгоритмов [1]. Компьютеры способны оперативно обрабатывать большие массивы данных, формы и характеристики объектов проектирования могут определяться варьированием параметров, входных данных, изменением алгоритмов. Компьютерные программы в данной конфигурации можно считать соавторами дизайнера, задачей которого является направление их в нужную сторону, корректировка и фильтрация результатов [2].

В применении генеративных технологий немаловажен фактор случайности, который позволяет получить огромное количество вариантов объектов и привести к неожиданным результатам за счет экспериментов, что придает особую значимость именно процессу разработки и позволяет говорить о различных методах и подходах к применению данных технологий.

Таким образом, специфика методов генеративного дизайна позволяет выделить два основных пути реализации дизайнерских задач [1, 2]:

1. Идея – Алгоритм – Результат;
2. Эксперимент – Идея – Результат.

Также можно выделить следующие основные подходы к реализации проектов в области генеративного дизайна [1]: построение алгоритмов; варьирование параметров вычислений; изменение результатов проектирования.

На сегодняшний день понятия «генеративный дизайн» и «генеративные технологии» являются очень широкими и неоднозначными. Нередко происходит переименование с терминами «параметрический дизайн», «вычислительный дизайн», которые имеют собственные особенности [3].

Необходимым средством для реализации проектов средствами генеративного дизайна является среда разработки, которая должна обладать возможностями для создания алгоритмов и позволять визуализировать данные.

Существуют специализированные средства для решения разноплановых дизайнерских задач генеративными методами в различных сферах. В настоящее время генеративные средства применяются, как на относительно примитивном уровне (например, автоматическое создание модульной сетки в веб-дизайне), так и в сложных разработках (например, при проектировании зданий, разработке дизайна мебели, обуви). Данные технологии и методы позволяют генерировать большое количество вариантов форм и графических решений, использование которых возможно и эффективно в брендинге, полиграфии, веб-дизайне.

Проанализировав функционал ряда программных продуктов области генеративного дизайна, можно выделить следующие возможности, позволяющие решать задачи графического дизайна:

- автоматическое создание макетов веб-сайтов, книжной и журнальной верстки. Группа первых технологий позволяет, например, исходя из контента статьи (количества абзацев, фотографий, их форматов, цитат и таблиц), выбирать типовой паттерн ее представления в эффектном журнальном виде, следить за тем, чтобы паттерны чередовались и материал выглядел разнообразно. Аналогичные возможности существуют и в веб. Это подбор стиля в зависимости от тематики, обработка фотографий, самостоятельный выбор лучших решений. Очевидно, что метод пригоден для решения стандартных задач, не требующих креативного подхода и новизны (браузерный движок Flipboard Duplo (JavaScript и CSS3), сервис The Grid, Netflix – вырезает персонажей, накладывает тексты);
- автоматический подбор цветовых сочетаний. В качестве входных данных используются изображения, фотографии, набор цветов. Программы генерируют на их основе гармоничные подборки, ориентируясь на образцы, заложенные разработчикам, а также на законы цветоведения. В частных случаях возможен, например, подбор цвета фона для размещения на нем графических элементов определенных цветов (Color Hunter, Generate it!, Color Explorer);
- генерация сочетаний шрифтов. Алгоритм анализирует символы, затем, используя базовые принципы сочетания шрифтов и исследуя имеющиеся в системе образцы грамотных решений, выдает собственные шрифтовые решения [4];
- стилизация изображений. Программы данного типа позволяют получать стилизованные изображения на основе полученного на входе, что значительно облегчает работу при необходимости, например, соблюдения единого стиля фотоматериалов в фирменной продукции или на сайте (Prisma, Mlvch);
- генерация 3D-объектов. Существуют достаточно мощные программы и алгоритмы, позволяющие, например, создавать 3D-объекты на основе скетчей, объединять различные формы и генерировать на их основе новые (Catia, StyLit, Grasshopper);
- решение частных задач и получение результатов с помощью экспериментов. Как было сказано, большая роль в генеративном искусстве отводится экспериментам. Зачастую их результаты являются довольно специфическими и узнаваемыми. Данная группа программ и алгоритмов позволяет создавать графические формы «с нуля» с помощью программного кода или задания количественных параметров. Характерным примером такой технологии является фрактальная графика. Изображения, получаемые с ее помощью, практически случайны и достаточно узнаваемы, но могут быть использованы в качестве элементов дизайна. В эту же категорию можно отнести параметрические шрифты (Aprophysis, ChaosPro);
- разработка динамической айдентики. Данная область применения технологий генеративного дизайна является наиболее перспективной и актуальной для дальнейшего исследования, может сочетать в себе использование комбинаций рассмотренных ранее технологий, а также алгоритмы, позволяющие реализовать визуальную реакцию форм объектов на звуки, наведение курсора и другие манипуляции. Таким образом, из графического элемента, полученного на входе, можно автоматически получить достаточное количество вариаций с анимированными переходами [5].

Главное отличие генеративных технологий такого типа от многочисленных шаблонных решений и сервисов-конструкторов сайтов, например, заключается в том, что данные

программы являются экспертными системами, т.е. имеют собственную «базу знаний» и могут частично выполнять функции «решателя». База знаний наполняется специалистами, в нее могут быть заложены, например, законы цветоведения и композиции, современные тенденции дизайна, качественные эталоны-образцы. Таким образом, программа анализирует собственную «базу знаний», входные параметры, заданные пользователем, и генерирует множество вариантов. Участие дизайнера (пользователя) минимально, от него требуется только правильная постановка задачи, выбор наилучшего решения и его доработка.

Таким образом, можно сформулировать основные направления деятельности генеративных систем:

- минимизация «рутинной», механической работы;
- расширение возможностей творческого поиска.

Однако результат выполнения тех или иных алгоритмов, за редким исключением, не может являться конечным продуктом и практически всегда требует доработки. Только при выполнении относительно простых функций генеративная система может являться «решателем», в остальных же случаях она является лишь «соавтором».

Стоит отметить, что совершенствование программного обеспечения и компьютерного оборудования позволяет достигать все более сложных визуальных эффектов, что придает особую значимость проблеме изучения новейших концепций и тенденций развития генеративных систем в области графического дизайна.

Литература

1. Генеративный дизайн: программирование как новый инструмент деятельности дизайнера [Электронный ресурс]. – Режим доступа: <https://www.scienceforum.ru/2015/921/11673>, своб.
2. Николаев И.П. Алгоритмы, фракталы, гармония // Pc week. – 2012. – № 14(380). – С. 31–36.
3. Свойства систем генеративного дизайна [Электронный ресурс]. – Режим доступа: <http://architecture.by/pogorelov/2015/03/generative-gesign-system-features>, своб.
4. Jon Gold. Taking The Robots To Design School, Part 1 [Электронный ресурс]. – Режим доступа: <http://www.jon.gold/2016/05/robot-design-school>, своб.
5. Процедурно-генерируемые логотипы в дизайне интерфейсов: технологии создания, возможности и ограничения использования [Электронный ресурс]. – Режим доступа: <http://samag.ru/archive/article/2910>, своб.



Башкирцева Екатерина Ивановна

Год рождения: 1995

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: b-ka95@mail.ru

УДК 004.92

ОБЗОР ОБЛАСТЕЙ ПРИМЕНЕНИЯ КОМПЬЮТЕРНОЙ ГРАФИКИ В РЕКЛАМЕ

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 областей применения компьютерной графики в рекламной сфере. Дано пояснение о широком диапазоне распространения компьютерной графики и об ее актуальности в современном мире. В ходе работы раскрыты основные понятия компьютерной графики и описаны современные тенденции развития компьютерной графики в рекламе.

Ключевые слова: компьютерная графика, реклама, визуализация, тенденция, область применения.

Компьютерная графика – это часть информатики, занимающаяся как созданием новых, так и обработкой уже существующих изображений [1].

Компьютерная графика используется во многих сферах деятельности:

- архитектура и инженерия (визуализация конечного вида строения или изделия);
- образование (создание обучающих тренажеров);
- медицина (построение объемных изображений по результатам различных медицинских исследований, управление визуализацией);
- развлечения (создание спецэффектов в кинематографе, game-design, мультфильмы);
- реклама (создание видеороликов, реклама в социальных сетях, типографика) и т.д.

По мере развития компьютерных технологий компьютерная графика приобрела новый статус и на данный момент используется везде, где необходимы создание и обработка изображений или цифровых данных.

Широкое применение компьютерной графики в рекламном деле объясняется тем фактом, что реклама в наше время стала главным звеном между потреблением и производством [2].

Актуальность темы работы обусловлена не только развитием компьютерной графики, но прежде всего развитием рекламной сферы, а также поиском новых способов привлечения покупателей. Использование в индустрии рекламы компьютерной графики играет важную роль в представлении продукта и его внешнем виде. Компьютерная графика позволяет расширить выразительные возможности, благодаря чему реклама получает высокую силу влияния на зрителя.

В индустрии рекламы наиболее популярной областью применения компьютерной графики является телевизионная реклама, т.е. рекламные ролики. Они способны наиболее полно отразить суть товара за незначительный промежуток времени. Качественный презентационный фильм (рекламный ролик) невозможен без компьютерной графики [3]. Среди визуальных эффектов остаются популярными использование покадровой анимации, спецэффектов, инфографики, совмещение видео с графикой, плавный переход между изображениями (морфинг). Новые тренды компьютерной графики в рекламе – использование высокого разрешения, графических сцен, неоновых цветов, поп-арта и панорамных видео.

Конкуренцию телевизионной рекламе составляет реклама в Интернете. В XXI веке большинство жителей планеты проводят время не за просмотром телевизора, а в социальных сетях. Онлайн-видео открывают перед рекламодателем практически безграничные возможности взаимодействия с аудиторией [4]. Одной из современных тенденций развития компьютерной графики в рекламе является создание мультимедийных записей, позволяющих пользователю инициировать заказ из видеоролика – shoppable video. Как правило, shoppable video демонстрирует сам продукт, а также ссылку на web-site поставщика, где потенциальный покупатель сможет узнать больше о товаре и совершить покупку.

Также современными тенденциями развития компьютерной графики в Интернет-рекламе являются:

- использование яркой цветовой палитры для расстановки акцентов или создания имиджа продавца;
- крупные заголовки, позволяющие быстро считать информацию о продукте;
- видео background или полноэкранные изображения, служащие для привлечения внимания покупателя или усиления визуального эффекта;
- минималистичный дизайн и использование геометрических фигур.

Правильное использование компьютерной графики при оформлении рекламы в социальных сетях позволит создать множество точек захвата, привлекающих внимание пользователя и побуждающих его углубиться в изучение продукта, на который направлена рекламная кампания.

Широкая область применения компьютерной графики в индустрии рекламы – печатная реклама. Популярной тенденцией развития компьютерной графики в печатной рекламе является использование шрифтового акцента. Он позволяет эффективно донести информацию в брошюре или буклете. Чем проще шрифт, тем лучше читается текст, это позволяет читателю сосредоточиться на содержании. Шрифтовой акцент позволяет выделить имя бренда или продукт. Этот визуальный прием позволяет дизайнеру влиять на восприятие изображения зрителем и создать визуальную взаимосвязь между элементами макета.

Привлечение внимания с помощью цветовых акцентов, необычного расположения элементов или различных текстур – это пример использования эмоциональной типографики, которая является одной из современных тенденций развития компьютерной графики в печатной рекламе. К ним также относятся:

- Material design – визуальный стиль, который считают одновременно природным и современным, вызывающим чувство реальности пользователя. Он отличается преднамеренным выбором цвета, масштабной типографикой и смелым использованием пустого пространства [5];
- HandMadeFont – уникальные шрифты для различных рекламных кампаний;
- использование отрицательного пространства для создания оригинального рекламного продукта;
- поп-арт натюрморты.

Таким образом, можно сделать вывод, компьютерная графика является неотъемлемой частью современного мира и с каждым днем она проникает в новые сферы общественной жизни [3]. Со временем появляются новые тенденции ее развития, в том числе в сфере рекламы.

При грамотном использовании компьютерной графики реклама позитивно воздействует на зрителя. Изучение применения компьютерной графики в рекламе является первым шагом на пути создания эффективного рекламного графического продукта.

Литература

1. Боресков А.В., Шикин Е.В. Компьютерная графика. Учебник и практикум для прикладного бакалавриата. – М.: Изд-во Юрайт, 2016. – 219 с.
2. Павелина М.М. Исследовательская работа «Реклама» // Материалы VIII Международной студенческой электронной научной конференции «Студенческий научный форум» [Электронный ресурс]. – Режим доступа: www.school-science.ru/2017/14/26521, своб.
3. Скворцов А.В. Использование компьютерной графики в создании рекламы // Научное общество Феникс [Электронный ресурс]. – Режим доступа: http://www.vfmgiu.ru/Nauchnoe-obschestvo-Feniks-1208/Nashi_publicacii/ispolzovanie_komp_grafiki_v_sozdanii_reklami/index.html, своб.
4. Пашутин Д. Реклама в онлайн-видео: основные тренды и тенденции [Электронный ресурс]. – Режим доступа: <http://www.adindex.ru/publication/analytics/forecasts/2014/03/24/108294.phtml>, своб.
5. Сиддикви Д. 7 трендов в графическом дизайне [Электронный ресурс]. – Режим доступа: <https://freelance.today/trendy/7-trendov-v-graficheskom-dizayne.html>, своб.

**Бидянова Екатерина Валерьевна**

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4207Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: xomka30098@gmail.com

УДК 004.051

**СПРАВНИТЕЛЬНЫЙ АНАЛИЗ РАБОТЫ БИБЛИОТЕК PROCESSING.JS, EASELJS,
LIBCANVAS С ОБЪЕКТОМ CANVAS****Бидянова Е.В.****Научный руководитель – к.педагог.н., доцент Государев И.Б.**

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе дан краткий обзор особенностей работы трех библиотек с объектом Canvas: Processing.js, EaselJS, LibCanvas.

Ключевые слова: HTML5, Processing.js, EaselJS, LibCanvas, Canvas.

В настоящее время при работе с HTML5 элементом Canvas [1] разработчики сталкиваются с проблемой производительности при создании графических элементов. Для решения этой и подобных проблем требуется использовать различные сторонние библиотеки, позволяющие оптимизировать работу с графикой.

Processing.js – это дочерний проект языка программирования Processing.

Основное преимущество Processing.js заключается в скорости работы, позволяя создавать сложную анимацию даже на медленных компьютерах. В Processing.js [2] также поддерживаются сложные эффекты, такие как затенение, освещение, 3D-трансформации.

Из недостатков можно выделить отсутствие возвращаемых объектов при их создании. Например, нарисовав на Canvas круг, Processing.js рисует круг и ничего не возвращает, в отличие от многих других библиотек. Для того чтобы изменить цвет нарисованного круга, необходимо нарисовать новый круг поверх существующего. Отсутствие объектов в данном случае обеспечивает скорость обработки сложных рисунков.

Также к недостаткам можно отнести необходимость изучения нового языка программирования Processing script, что может усложнить процесс разработки.

EaselJS – open-source JavaScript библиотека, основанная на библиотеке Create.js, упрощающая работу с Canvas при создании игр и др. [3].

В отличие от Processing.js EaselJS поддерживает объекты, что означает, что при создании на объекте Canvas круга, вернется объект круга, у которого можно изменять свойства, добавлять события, что позволяет добавить интерактивности. Но подобное хранение объектов ведет за собой увеличение скорости выполнения программ.

LibCanvas – фреймворк для работы с объектом Canvas, построенный на базе AtomJS [4].

Из основных недостатков LibCanvas можно выделить высокий порог вхождения и отсутствие широкого сообщества разработчиков, что делает решение возникающих проблем более сложным. При этом у LibCanvas широкий функционал, в основном направленный на создание игр.

Сравнение Processing.js, EaselJS и LibCanvas было проведено на основании эффективности кода. Для этого была поставлена простейшая задача: нарисовать три прямоугольника (рис. 1).

Simple Canvas example

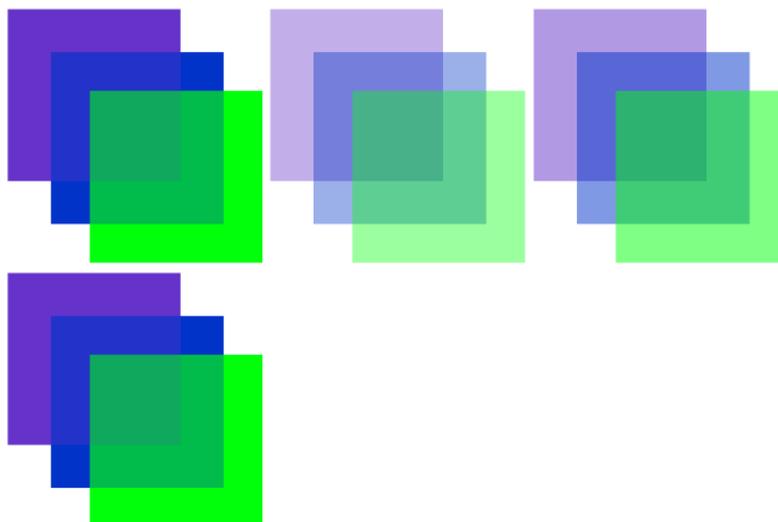


Рис. 1. Native JavaScript, Processing.js, EaselJs, LibCanvas

Тестирование проводилось с помощью JSPerf, веб-приложение для создания тестов для проверки js-скриптов на эффективность.

Ниже представлен исходный код приложения.

```
1 <script src = "https://cloud.github.com/downloads/processing-js/processing-js/processing-1.4.1.min.js"></script>
2 <script src = "https://libcanvas.github.io/files/js/atom.js"></script>
3 <script src = "https://libcanvas.github.io/files/js/libcanvas.js"></script>
4 <script src = "https://code.createjs.com/easeljs-0.8.2.min.js"></script>
5 <h1>Simple Canvas example</h1>
6 <canvas id="canvas_processing" width="300" height="300" ></canvas>
7 <canvas id="canvas_easel" width="300" height="300" ></canvas>
8 <canvas id="canvas_libcanvas" width="300" height="300" ></canvas>
9 </script>
10 //Processing.js
11
12 function ffff(processingInstance){
13 with (processingInstance) {
14 size(300, 300);
15 background(255, 255, 255);
16 noStroke();
17 // ProgramCodeGoesHere
18 fill(100,50,200,100);
19 rect(5, 5, 200, 200);
20 fill(0,50,200,100);
21 rect(55, 55, 200, 200);
22 fill(0,255,10,100);
23 rect(100, 100, 200, 200);
24
25 };
26 }
27 var processingCanvas = document.getElementById("canvas_processing");
28 var processingInstance = new Processing(processingCanvas);
29
```

```

30
31 //EaselJS
32
33 var stage = new createjs.Stage('canvas_easel');
34 rect_1 = new createjs.Shape();
35 rect_2 = new createjs.Shape();
36 rect_3 = new createjs.Shape();
37 rect_1.graphics.beginFill("rgba(100,50,200,0.5)").drawRect(5, 5, 200, 200);
38 rect_2.graphics.beginFill("rgba(0,50,200,0.5)").drawRect(55, 55, 200, 200);
39 rect_3.graphics.beginFill("rgba(0,255,10,0.5)").drawRect(100, 100, 200, 200);
40 stage.addChild(rect_1, rect_2, rect_3);
41
42 //LibCanvas
43
44 LibCanvas.extract();
45 var canvas_lib = document.getElementById("canvas_libcanvas");
46 var context = canvas_lib.getContext('2d-libcanvas');
47
48 </script>
49
50

```

Полученные результаты приведены на рис. 2.

Testing in Chrome 55.0.2883 / Windows 10 0.0.0		
	Test	Ops/sec
Processing.js	ffff(processingInstance);	10,023 ±1.25% 88% slower
EaselJS	stage.update();	39,780 ±0.97% 54% slower
LibCanvas	context .fill(new Rectangle(5, 5, 200, 200), "rgba(100,50,200,0.5)") .fill(new Rectangle(55, 55, 200, 200), "rgba(0,50,200,0.5)") .fill(new Rectangle(100, 100, 200, 200), "rgba(0,255,10,0.5)");	29,499 ±1.17% 66% slower

Рис. 2. Полученные результаты

Исходя из полученных результатов, можно прийти к выводу, что на основании поставленной задачи, производительность веб-приложения, написанного с использованием EaselJS, выше, чем с использованием других библиотек.

Тем не менее, Processing.js, EaselJS и LibCanvas обладают рядом преимуществ, при выборе библиотеки для более сложных проектов, необходимо отталкиваться от поставленных задач.

Литература

1. David Flanagan. Canvas Pocket Reference: Scripted Graphics for HTML 5. – O'Reilly Media, 2010. – 112 p.
2. [Электронный ресурс]. – Режим доступа: <http://processingjs.org/>, своб.
3. [Электронный ресурс]. – Режим доступа: <http://www.createjs.com/easeljs>, своб.
4. [Электронный ресурс]. – Режим доступа: <https://github.com/theshock/libcanvas>, своб.



Бобрик Ольга Александровна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: bobrik_olga94@mail.ru

УДК 004.9

ПЕРСПЕКТИВНЫЕ НАПРАВЛЕНИЯ РАЗВИТИЯ ИНФОГРАФИКИ

Бобрик О.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 проанализированы работы победителей конкурсов Malofej 2016 и Kantar Information Is Beautiful 2016, на их основе и на основе ряда публикаций выделены перспективные направления в развитии инфографики.

Ключевые слова: инфографика, тенденции инфографики, Malofej 24, Kantar Information Is Beautiful 2016, визуализация.

В работе на основе анализа работ победителей известных конкурсов в мире графики «Malofej 24» 2016 [1] и «Kantar Information Is Beautiful» 2016 [2] и на основе ряда публикаций [3, 4] выделены направления инфографики, которые сейчас популярны и активно развиваются. Результат анализа объясняет актуальность настоящей работы, так как определяемые тренды являются современными и перспективными. **Целью исследования** являлось выделение перспективных тенденций современной инфографики. Исходя из цели, необходимо было решить следующие задачи: выбрать пул конкурсных работ; выделить приемы, используемые для создания инфографики; проанализировать полученные тенденции; выбрать перспективные направления. Также для реализации цели были использованы такие методы, как: изучение и обобщение, сравнение, анализ, описательный метод.

В работе С.И. Симаковой [3] отмечается, что в начале XXI века инфографика развивалась большими шагами, особенно в области рекламы и промоушена. Однако из-за этого ее информационно-аналитические возможности были отодвинуты на второй план, и «произошло перенасыщение медиапространства инфографикой, часто очень низкокачественной, исчез индивидуальный подход к каждому проекту» [3].

В 2014 году инфографика начала преодолевать этап кризиса. Возглавили это восстановление Росс Крукс (Ross Crooks), сооснователь агентства Five Column Media, Брэд МакКарти (Brad McCarthy), онлайн-ресурс Visual.ly и другие специализированные онлайн-издания [3]. Стали возникать новые научные работы по визуализации, отслеживаться новые тенденции в статистике и дизайне, бренды и компании стали работать над визуальной коммуникацией. Сабрина Илдер, UX-специалист и основатель UXkids, считает, что перейти на новый уровень инфографике помогли следующие тренды: мобильность устройств; начало периода адаптивного дизайна; интерактивность; акцент на сторителлинг [4]. Современная инфографика становится мультимедийной: графики обогащаются текстом, фотографиями, медиа- и аудиофайлами. Возникают новые тренды: видеоинфографика, анимированная, интерактивная [3].

Для того чтобы подтвердить эти тренды и выделить другие, авторами проанализированы работы победителей двух международных конкурсов в области инфографики.

Международный конкурс «Malofej Awards» проводится ежегодно с 1993 г. в Памплоне (Испания) некоммерческой организацией – «The Society for News Design» («SND»). В 2016 году конкурс проводился 24-й раз, было прислано 1388 работ из 150 СМИ 33 стран мира [1]. Участники разделяются на печатную и онлайн-графику и еще на ряд подкатегорий в этих группах. Для анализа была выбрана только онлайн-графика, которая представлена в свободном доступе на сайте конкурса.

Конкурс «Kantar Information is Beautiful Awards» создан в 2012 году журналистом и дизайнером студии «Information is Beautiful» Дэвидом МакКэндлсом (David McCandless) и Азизом Ками (Aziz Cami), в то время креативным директором «Kantar». Принять участие в конкурсе может любой желающий, предварительно оплатив взнос [2]. Для анализа были выбраны работы победителей из категорий инфографика и лучшая инфографика на неанглийском языке.

Общее количество проанализированных работ – 55. Среди них были выделены следующие элементы, используемые в инфографике:

1. анимация и интерактивность;
2. сторителлинг;
3. скроллинг;
4. видео, аудио;
5. использование больших данных.

Анимация и интерактивность. По мнению С.И. Симаковой «среди наиболее показательных и важных трендов главным остается интерактивная инфографика» [3]. В своей работе [3] Симакова отмечает, что интерактивность может изменяться от простой анимации до уровня индивидуального взаимодействия с каждым пользователем и фильтрацией данных. К первому виду относятся анимированные элементы, например подсвечивание выделяемых пользователем объектов, появление информации и различных диаграмм и т.п. Такие приемы хоть и не удивляют уже людей, но облегчают процесс создания инфографики, ее чтение и понимание. Интерактивность второго вида встречается реже и является наиболее перспективной. Пользователь самостоятельно выбирает необходимые сведения – «инфографика не только существует вместе с ним, но и напрямую зависит от него» [3]. Оба вида интерактивности повышают читаемость, привлекательность и общее качество инфографики. Из проанализированных работ 32 были анимированные, а в 12 присутствовала интерактивность в качестве взаимосвязи действий пользователя и выводимой информации.

Сторителлинг. Также можно встретить такое понятие как визуальный сторителлинг. Это прием влияния на аудиторию через рассказывание историй. Сабрина Илдер, в работе «Popular Methods for Online Storytelling» [4] отмечает, что эффективность изображений для передачи историй превосходит текст, так как их наш мозг обрабатывает быстрее в 60 тысяч раз. Следовательно, инфографика – это прекрасный вариант сторителлинга. Такие картинки популярны у пользователей из-за своей емкости и минимальности использования текста [4]. Из проанализированных работ 25 имели формат визуального повествования, что подтверждает популярность такого приема.

Скроллинг. Так как рассматривались работы в категории онлайн-графики, одним из элементов стал скроллинг – это способ навигации, при котором информация перемещается в вертикальном или горизонтальном направлении. Эксперты студии «LPgenerator» [5] отмечают, что стиль длинного скроллинга, в котором разные части и понятия складываются в непрерывный поток, делает его очень выгодным для инфографики. Такой формат позволяет разбить большие порции информации на более мелкие и легкоусвояемые [5]. В проанализированных работах 11 использовали скроллинг для изменения информации, 30

использовали скроллинг в качестве навигации, 7 использовали кнопки для навигации, остальные 5 изображений были статичными.

Видео и аудио. Видеоинфографика является подвидом динамичной инфографики и популярна в качестве рекламных и корпоративных роликов, презентаций. Одна конкурсная работа «Blue whale: The Largest animal on Earth» была выполнена в качестве VR (виртуальная реальность) приложения для Android. Из представленных конкурсных работ 8 использовали видео для отображения инфографики, и в 3 применялось аудио для лучшего восприятия информации.

Использование больших данных. Про тренд сближения инфографики с большими данными Симакова отмечает следующее: «Последние (большие данные) всегда отличались комплексностью и сложностью, и без подготовки с ними работать было довольно сложно [3]. Визуализация же решит основные проблемы восприятия больших данных и сделает их доступными для широких масс. Поэтому тенденция к сближению инфографики и больших данных считается наиболее перспективной для обеих сфер» [3]. Данную тенденцию отмечает и дизайнер инфографики Эрнесто Оливарес (Ernesto Olivares): «Что будет, если смешать феноменальную способность инфографики рассказывать уникальную историю с глубоким аналитическим потенциалом больших данных? Звучит как самая дикая мечта медийного работника, но мы уже можем это наблюдать» [3]. Так как визуально достаточно сложно определить, сколько данных было использовано при создании той или иной инфографики, авторы решили не выделять какие-либо работы по этому признаку. Однако все же стоит отметить, что практически все работы представляют собой результаты огромных трудов по анализу и обработке информации.

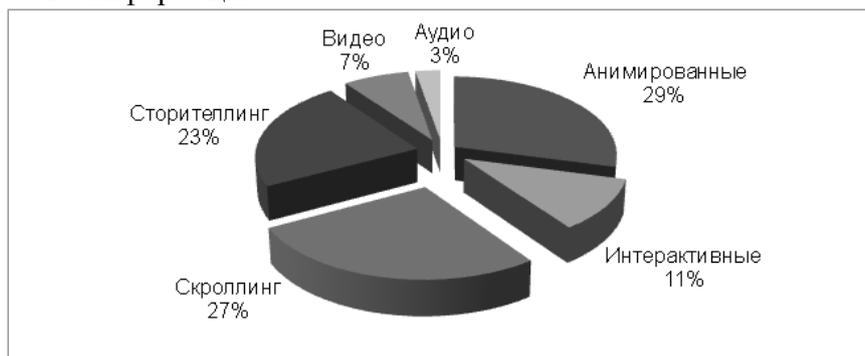


Рисунок. Диаграмма результатов анализа работ (количество в %) по рассмотренным направлениям инфографики

В заключение работы представлена диаграмма результатов анализа работ по выделенным направлениям (рисунок). По диаграмме можно увидеть, что больше всего в работах используют анимацию, на втором месте инфографика со скроллингом, на третьем месте инфографика, передающая информацию в формате сторителлинга. Также набирают популярность работы, использующие интерактивные элементы в своих проектах, и растет количество информации, которую необходимо обработать для создания более точной, правдивой и интересной инфографики. В дальнейшем планируется более детально проанализировать одно из выделенных направлений, а также создать собственный проект по этой тематике.

Литература

1. Malofejj 24: Winners. – Режим доступа: <http://www.snd-e.com/themed/snde/files/docs/249/065/m24awardslistok.pdf>, своб.
2. Kantar Information Is Beautiful Awards // Kantar [Электронный ресурс]. – Режим доступа: <http://www.informationisbeautifulawards.com/news/188-2016-the-winners>, своб.
3. Симакова С.И., Федотовский В.В. Инфографика: прошлое, настоящее, будущее // Знак: проблемное поле медиаобразования: науч. журн. – 2016. – № 3(20). – С. 13–25.
4. Pider S. Popular methods for online storytelling [Электронный ресурс]. – Режим доступа: <http://sixrevisions.com/user-experience-ux/online-storytelling-methods>, своб.

5. LPgenerator: размышляем на тему длинной прокрутки, или так называемого скроллинга [Электронный ресурс]. – Режим доступа: <http://lpgenerator.ru/blog/2016/04/04/razmyshlyaem-na-temu-dlinnoj-prokrutki-ili-tak-nazyvaemogo-skrollinga>, своб.



Богданов Дамир Рашидович

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4208

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: damir.bogdanov@niuitmo.ru



Шуклин Дмитрий Анатольевич

Год рождения: 1979

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, к.педагог.н.

e-mail: kpd@limtu.ru

УДК 004.43

СРАВНЕНИЕ ОСНОВНЫХ КЛИЕНТСКИХ JAVASCRIPT ФРЕЙМВОРКОВ

Богданов Д.Р.

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

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе представлен сравнительный анализ популярных JavaScript фреймворков – Backbone.js, Ember.js, AngularJS – по ряду важных критериев, к числу которых относятся функциональность, продуктивность разработки, сообщество разработчиков, защита от утечки и т.д.

Ключевые слова: JavaScript, Backbone.js, Ember.js, AngularJS, фреймворк.

Одним из инструментов организации динамических веб-страниц является прототипно-ориентированный сценарный язык программирования JavaScript, развитие и утверждение новых стандартов которого привело к появлению значительного количества написанных на его базе библиотек и программных платформ – фреймворков. Разработчики фреймворков с каждым годом увеличивают функциональность своего продукта, скорость его работы и продуктивность, успешно борются с такими проблемами, как утечка памяти и малая гибкость фреймворка, детальнее документируют свой продукт [1]. При этом возникает проблема выбора фреймворка для оптимизации клиентского JavaScript приложения, что актуализирует тему настоящего исследования и ставит перед автором ряд задач, к числу которых относится определение критериев сравнения и анализ основных. JavaScript фреймворков в контексте соответствия данным критериям.

Обзор литературы позволил автору выделить следующие критерии для сравнения фреймворков: гибкость, функциональность (наблюдаемость, маршрутизация, представление связываний, двунаправленное связывание, вложенные представления, отобранный список

просмотра), продуктивность разработки, сообщество разработчиков, инфраструктура, защита от утечек памяти [2].

Для сравнительного анализа по названным критериям были выбраны следующие фреймворки – Backbone.js, Ember.js, AngularJS. Данный выбор обусловлен в первую очередь такими факторами, как большое сообщество (более 2 тысяч человек) и наличие подробной документации. Сравнение названных фреймворков было проведено на основе личного профессионального опыта работы автора, анализа литературы и интернет-источников, изучения мнений конечных пользователей исследуемых приложений.

Критерий гибкости был оценен с позиции возможности интеграции специализированных плагинов и библиотек с фреймворком. Лучшим оказался Backbone.js, минимализм которого обеспечил гибкость в плане архитектуры веб-приложения (модели, коллекции и представления являются строительными блоками веб-приложения) [3]. Ember.js и AngularJS – являются до некоторой степени гибкими, но можно столкнуться с очень жесткими подходами к решению определенного круга задач проектирования. Функциональность является важнейшим качеством и технологическим критерием выбора фреймворка. Ранее уже были названы составляющие данного параметра сравнения. Стоит отметить, что всем показателям соответствует AngularJS и Ember.js. Однако используя Backbone.js, можно реализовать большую часть этих функций вручную или через плагины. Анализ продуктивности разработки позволил сделать следующие выводы. Backbone.js требует написания очень большого объема шаблонного кода, что совершенно ненужно и оказывается прямой угрозой производительности труда. AngularJS не хватает достаточного объема соглашений для быстрой разработки кода. Ember.js с точки зрения продуктивности обходит все фреймворки, так как представляет собой список соглашений, автоматизирующих большое число функций. Backbone.js имеет самую развитую инфраструктуру, самое большое количество дополнительных библиотек и плагинов. Инфраструктура AngularJS менее развита, чем у Backbone.js. Ember.js имеет несколько дополнительных библиотек, но в целом инфраструктура не развита. AngularJS и Ember.js имеют встроенную систему защиты от утечек памяти. Backbone.js требует делать эту работу самостоятельно, своевременно удалять обработчики событий. На основе проведенного анализа автором была составлена сводная таблица.

Таблица. Сравнение JavaScript фреймворков

Показатели	Backbone.js	Ember.js	AngularJS
Гибкость	5	3	3
Функциональность	2	5	5
Продуктивность разработки	2	5	4
Сообщество разработчиков	4	4	5
Инфраструктура	5	3	4
Защита от утечек памяти	3	5	5
Итого:	21	25	26

Таким образом, можно утверждать, что рассматриваемые фреймворки имеют примерно одинаковый достаточный уровень набора инструментов для реализации клиентского приложения. Каждый фреймворк имеет свою собственную идеологию структурирования кода в приложении, подходы к работе с данными пользователей, шаблонизацию и прочие методы организации работы приложения.

Литература

1. Богатырев В.В., Сафина В.К. Современные средства веб-разработки // Материалы Всероссийской научно-практической конференции «Информационно-телекоммуникационные системы и технологии». – 2015. – С. 162.

2. Буренин С.Н. Web-программирование и базы данных: учебный практикум. – М.: Изд-во Московского гуманитарного университета, 2014. – 120 с.
3. Османи Э. Разработка Backbone.js приложений. – СПб.: Питер, 2014. – 352 с.

**Букирёва Людмила Александровна**

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107
Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: bukirevala@gmail.com

**Государев Илья Борисович**Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна,
к.педагог.н., доцент

e-mail: ilia.gossoudarev@gmail.com

УДК 004.5

**ПРОТОТИПИРОВАНИЕ КАК НЕОТЪЕМЛЕМЫЙ ЭТАП ПРОЦЕССА СОЗДАНИЯ
САЙТА****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 рассмотрен процесс проектирования сайтов, изучены этапы разработки пользовательского интерфейса, проанализированы различные подходы к созданию прототипов сайтов, а также особенности существующих методов создания прототипов.

Ключевые слова: wireframes, дизайн, прототип, интерактивный прототип, прототипирование.

Одним из ключевых процессов разработки веб-продукта является проектирование пользовательского интерфейса, которое можно разделить на следующие этапы [1]: предпроектный анализ; оценка спланированной работы; создание структурных схем страниц (wireframes); разработка дизайна; создание интерактивного прототипа системы. Последние три этапа образуют процесс прототипирования, под которым понимается разработка прототипа сайта.

На сегодняшний день нет однозначного мнения в вопросе необходимости этапа прототипирования. Наряду со специалистами, которые отмечают исключительно положительные стороны внедрения данного процесса в разработку сайта, существуют мнения и о том, что создание прототипа не всегда является целесообразным выбором. В связи с этим **целью работы** явился анализ процесса прототипирования сайтов.

Прототип в веб-дизайне – это страницы будущего сайта в виде эскиза или HTML-документа, где отображены структурные элементы будущего сайта: меню, кнопки, формы и др. Прототип может быть статичным или динамичным.

На стадии создания структурных схем страниц (wireframes), прототип является статичным и представляет собой «скелет» будущего интерфейса с размещением основных функциональных элементов. Такие прототипы дают первое наглядное представление о будущей системе, позволяют поставить задачи дизайнеру, верстальщику интерактивного прототипа, разработчикам [2].

После разработки дизайна при необходимости происходит создание интерактивного прототипа. Это действующая модель пользовательского интерфейса: в него включены основные страницы и процессы работы системы. Несмотря на то, что это только имитация работы – данные не сохраняются, и отсутствует работа с сервером – прототип позволяет без прочтения документации понять, как работает система [3].

Как правило, интерактивный прототип представляет собой соединенные между собой HTML-страницы с имитацией реакции системы через JavaScript.

Существует два подхода к созданию интерактивных прототипов [4]: черно-белый прототип на основе структурных схем страниц и цветной прототип, основанный на принятом заказчиком визуальном дизайне системы.

Получить черно-белый прототип можно уже на раннем этапе проектирования, не дожидаясь отрисовки визуального дизайна. Это дает возможность, как можно раньше начать юзабилити-тестирование. Но такой прототип в редких случаях подойдет для демонстрации, поскольку, как правило, заказчику сложно абстрагироваться от прототипа, состоящего главным образом из «функционала». В связи с этим применение данного вида прототипа окажется наиболее уместным для взаимодействия дизайнера и разработчика.

Разрабатывать черно-белый прототип стоит в тех случаях, когда либо бюджет проекта очень маленький, достаточный только на создание простейшей модели, либо наоборот, когда времени и бюджета достаточно на создание черного-белого и цветного прототипов.

Гораздо проще общаться с заказчиком, имея цветной прототип, поскольку наличие дизайна улучшает восприятие прототипа. Юзабилити-тестирование также пройдет более эффективно, и его результат будет ближе к реальности.

Однако делать цветной прототип может быть достаточно дорого и долго, поскольку такие прототипы в основном создаются вручную, с помощью обычных программ для HTML-верстки.

Обобщенные и систематизированные причины использования и отказа от использования прототипа представлены в таблице.

Таблица. Причины использования прототипа

№	Причины использовать прототип	Причины отказаться от прототипа
1	может выступать в роли средства быстрого согласования проекта, поскольку имеет визуальное представление интерфейса разрабатываемого продукта	описывает интерфейсы сайта – серверную часть программирования и базы данных все равно необходимо описывать в техническом задании
2	позволяет сэкономить время для последующих этапов создания продукта: внесение изменений на стадии прототипирования занимает меньше времени, чем на стадии дизайна или разработки	для восприятия даже глубоко проработанного прототипа необходимо абстрактное мышление – заказчики-«визуалы» иногда не воспринимают прототип и могут работать только с дизайном
3	снижает стоимость ошибки, так как устранение ошибки на этапе прототипирования обходится дешевле, чем на завершающих этапах разработки, а тем более после ввода в эксплуатацию	практически неактуален для простых проектов, которые состоят из одного-двух шаблонов и заполняются однотипным содержанием

№	Причины использовать прототип	Причины отказаться от прототипа
4	при отсутствии прототипа вся работа по проектированию переходит на этап дизайна, а проектирование при помощи неспециализированных средств может привести к дополнительным временным и денежным затратам или оказаться бесперспективным	если проект делается на уже существующей технологической платформе, и ее функции заведомо ограничены, прототип не нужен, поскольку функционал уже определен
5	упрощает следующий за ним этап дизайна и делает его более предсказуемым: наличие схем страниц сайта с размещением основных элементов позволяет проще и дешевле отрисовать макеты	не поможет в работе над проектом, требующим креативного визуального решения – сложная графика и идеи лучше демонстрировать в форме презентации и эскизов, но не прототипа
6	является готовым проектом, с которым при необходимости можно обратиться в любую компанию для продолжения работы	если делается редизайн веб-проекта с сохранением функционала, роль прототипа выполняет он сам – делать прототип заново обычно нет необходимости
7	при наличии интерактивного прототипа у заказчика есть возможность увидеть сайт и даже начать его тестировать на самых ранних этапах, еще до того, как начнутся дорогие производственные работы	детальный прототип всего проекта может стать бесполезной работой, если проект планируется вести по гибким методологиям разработки
8	при создании прототипа заказчику становится проще формулировать требования к проекту, а исполнителю – вносить корректировки	
9	поскольку в интерактивном прототипе демонстрируется работа интерактивных элементов, а также имитируется работа серверной части, есть возможность отрабатывать на нем сложные этапы взаимодействия с посетителем	
10	может использоваться в качестве презентации для заказчиков и инвесторов, поскольку демонстрация прототипа создаваемого сайта является лучшим способом показать планируемый проект	
11	позволяет вовремя определить, что проект является слишком дорогим или нереализуемым	

В завершении проведенного анализа процесса прототипирования, можно сделать вывод о том, что использование прототипа при разработке сайта необходимо, однако есть причины ограничить масштабы его применения. К каждому проекту требуется индивидуальный подход, поскольку выбор средства для прототипирования напрямую зависит от размера проекта, его сложности, бюджета, сроков и подготовки заказчика. В разных ситуациях нужны (и возможны) различные подходы, но очевидно, что прототип – это гибкий рабочий инструмент, который в том или ином виде необходим в процессе создания сайта.

Литература

1. Проектирование пользовательских интерфейсов. Краткий обзор процесса [Электронный ресурс]. – Режим доступа:

<http://web.archive.org/web/20080112171847/http://www.jvetrau.com/2007/12/18/proektirovanie-polzovatel'skih-interfejsov-kratkiy-obzor-protssessa>, своб.

2. Корзина М.И., Поташова М.А., Лысенко А.А. и др. Дизайн веб-сайта на основе информационной модели системного проектирования // Дизайн, бизнес и общество. Грани взаимодействия: материалы междунаро. научной конф. – 2012. – С. 28–30.
3. Лысенко В.А., Лысенко А.А., Корзина М.И. Системное проектирование (дизайн) web-сайта // Дизайн. Материалы. Технология. – 2013. – № 1(26). – С. 116–122.
4. Ротина К.М. Анализ этапов проектирования сайтов // Вестник МГУП. – 2013. – № 9 [Электронный ресурс]. – Режим доступа: <http://cyberleninka.ru/article/n/analiz-etapov-proektirovaniya-saytov>, своб.



Ванина Наталья Валерьевна

Год рождения: 1995

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4106

Направление подготовки: 09.04.02 – Информационные системы
и технологии

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

УДК 004.925.5

ЦВЕТОВАЯ СХЕМА WEB-СТРАНИЦЫ ДЛЯ ЛЮДЕЙ С НАРУШЕНИЕМ ЦВЕТОВОСПРИЯТИЯ

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 восприятия цвета у людей страдающих дальтонизмом, выявлены некоторые способы повышения юзабилити сайта для людей с нарушением цветовосприятия, составлен перечень рекомендаций при создании дизайна web-страниц, приведены и проанализированы примеры адаптации цветовой схемы интерфейса, выявлены достоинства и недостатки данных рекомендаций и примеров.

Ключевые слова: дальтонизм, цветовая схема, интерфейс, дизайн web-страницы, юзабилити web-страницы.

С ростом популярности web-пространства появляется потребность в адаптации интерфейса для пользователей с любыми видами отклонения восприятия визуальной или аудиальной информации на сайте. Одной из проблем, возникающих при разработке дизайна (визуальной составляющей) web-страницы, является адаптация цветовой схемы графических элементов для людей, страдающих дальтонизмом.

Дальтонизм – слепота на красно-зеленый цвет (протанотопия – протаномалия, дейтранотопия – дейтраномалия), реже – сине-желтый цвет (тританотопия – тританомалия); совсем редко встречаются люди с полной цветовой слепотой (ахроматопсия). По недавним исследованиям примерно 8% мужчин и 0,5% женщин (4,5% населения) страдают данным отклонением [1]. Исследования, касающиеся популярности видов дальтонизма, приведены на рис. 1. Также данный рисунок наглядно показывает, как именно люди с цветовыми отклонениями воспринимают цвета [2].

Проанализировав исследования в области восприятия цветовых схем различных интерфейсов дальтониками, был составлен список рекомендаций, которые могут быть полезными при разработке дизайна web-страниц.

	Normal Vision	Protanotopia	Deuteranotopia	Tritanotopia
Men	91.4%	2.45%	6.1%	0.011%
Women	99.6%	0.04%	0.36%	0.04%
Overall	95.5%	1.25%	3.24%	0.025%

Рис. 1. Виды дальтонизма, схема восприятия цвета у дальтоников

1. Использовать безопасную палитру цветов. Безопасная палитра состоит из 216 цветов шестнадцатеричной системы, каждый оттенок должен содержать в своем коде значение 00, 33, 66, 99, CC или FF. Если в сочетании используются другие числа и буквы, цвет является «опасным» и будет иначе восприниматься людьми с нарушением цветовосприятия (Пример безопасных цветов: #33;33;CC, #00;FF;99, #FF;99;CC) [3]. Преимущество данного способа адаптации – безопасность отображения на любых мониторах и в любых браузерах. Недостаток – ограниченность в цветовом разнообразии дизайна.
2. Нежелательные сочетания. При разработке элементов следует избегать сочетания красного и зеленого в отображении интерактивных элементов страницы. Как уже было замечено ранее, красно-зеленая слепота является самой распространенной среди дальтоников. Один из вариантов решения данной проблемы – замена сочетания красный–зеленый на красный–синий. Также стоит отказаться от таких сочетаний как: зеленый–синий, зеленый–черный, зеленый–коричневый, зеленый–серый, салатный–желтый, синий–серый, синий–фиолетовый [4]. Положительные и отрицательные стороны данного способа схожи с предыдущим решением проблемы.
3. Контрастные оттенки и цвета. Следует обратить внимание на текстовые ссылки и активные элементы. Подсвечивание объектов нейтральными цветами, избегание оттеночной передачи выбранного элемента, сделает интерфейс более адаптивным и позволит избежать трудностей в пользовании сайтом. На рис. 2 показан способ отображения активных элементов. Недостатком данного метода можно считать излишнюю яркость цветовой палитры, что может быть неудобным для отображения неактивной графической информации и дизайна в целом.

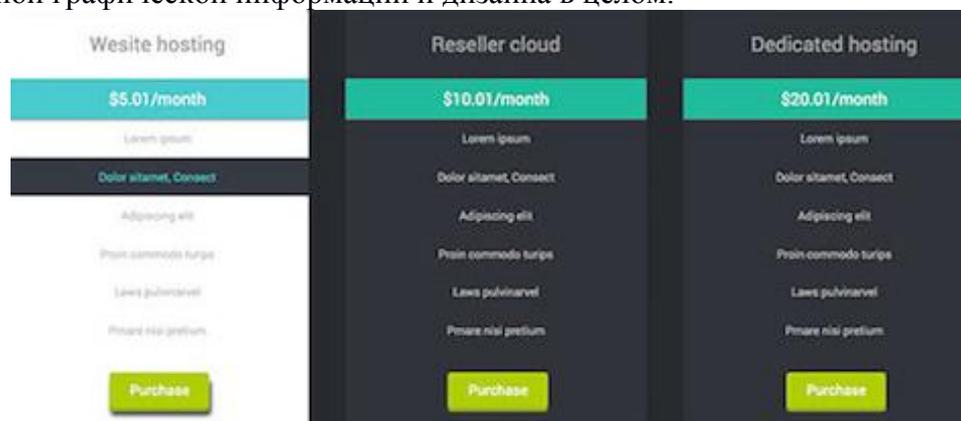


Рис. 2. Пример контраста выбранных элементов и фона

4. Комбинация цвета и символов. При отображении индикатора состояния объекта не стоит ориентироваться только на цветовую информативность. В большинстве случаев смысл

можно передать с помощью универсальных символов. Примером является панель регистрации на сайте facebook.com, представленная на рис. 3 [4]. Трудности в данном случае могут возникнуть в отсутствии ситуативных символов, а также в объектной перегрузке интерфейса.

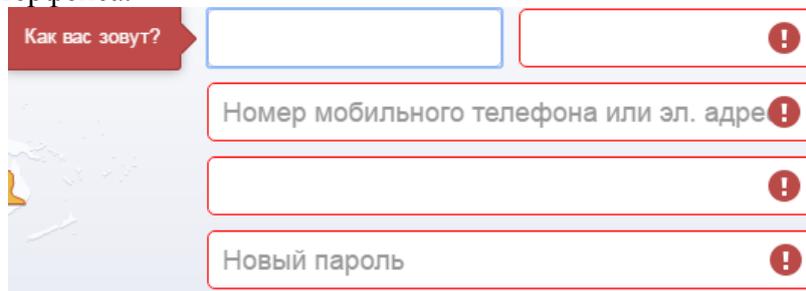


Рис. 3. Пример символического обозначения состояния

5. Узор и текстурирование. Если существует необходимость сохранить цветовую схему, выделение элементов можно осуществить путем текстурирования или нанесение узора на объект. Пример сохранения цветовой схемы с помощью узора представлен на рис. 4 в игре Two Dots.

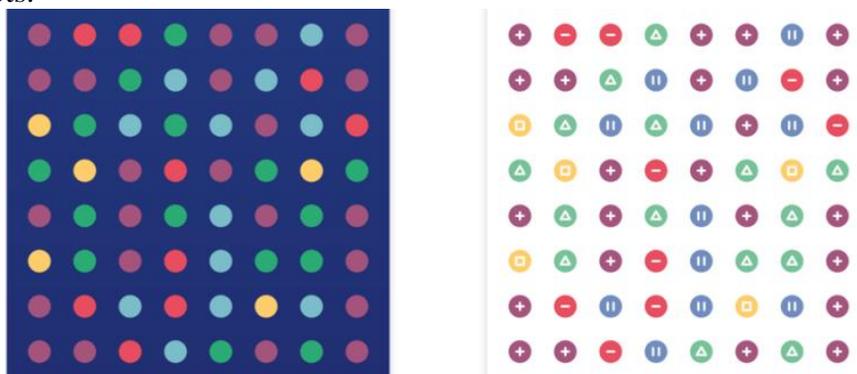


Рис. 4. Пример контрастного узора в игре Two Dots

Данный способ имеет свою специфику и трудности реализации, в основном применяется в играх, редко – в графиках и диаграммах.

6. Минималистичный дизайн. Минимализм в дизайне позволяет ограничить цветовую палитру, лаконично и доступно представить всю необходимую информацию и сделать интерфейс дружелюбным для пользователей с различными видами цветовой слепоты. Хотя данный способ носит узкую направленность, основные тенденции настроены на развитие данного направления дизайна, что может стать решением поставленной проблемы. Пример минималистичного дизайна представлен на рис. 5.

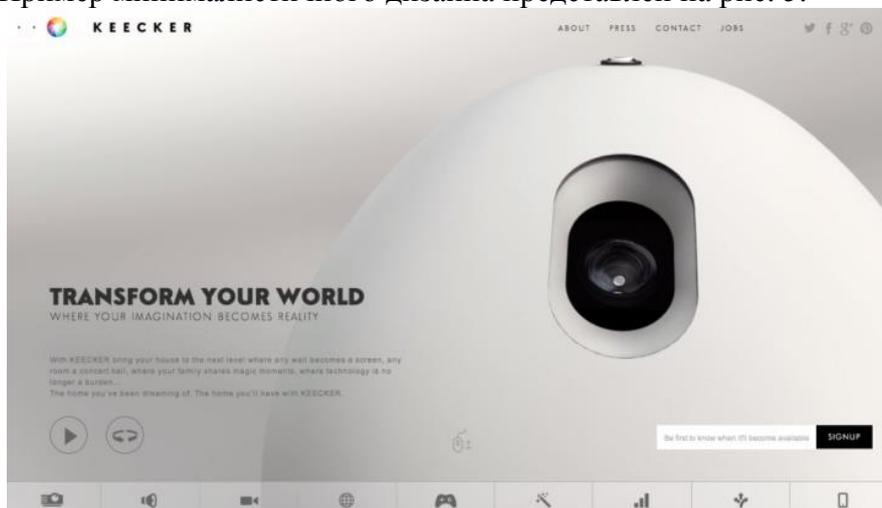


Рис. 5. Сайт в минималистичном стиле

В работе представлено краткое описание особенностей цветовосприятия дальтоников. На основе изученных данных и уже ранее проведенных исследований составлены рекомендации для создания дизайна наиболее адаптивного под все виды нарушения цветовосприятия, выделены отрицательные и положительные характеристики каждого из способов. Хотя данный перечень состоит из исчерпывающих методов создания максимально дружелюбного интерфейса и решает все поставленные проблемы, встает вопрос о сохранении дизайна web-страниц, что на данный момент остается приоритетным при создании сайтов.

В ходе дальнейшего исследования планируется изучить уже существующие технологии адаптации цветовой схемы web-страницы для дальтоников с сохранением изначального дизайна, проанализировать и выявить слабые и сильные стороны данных программ. На основе полученных данных, с помощью современных инструментов программирования, создать универсальный интегрируемый код, позволяющий адаптировать цветовую схему любого сайта [5].

Литература

1. The National Eye Institute (NEI). Facts About Color Blindness [Электронный ресурс]. – Режим доступа: https://nei.nih.gov/health/color_blindness/facts_about, своб.
2. Miller M. Great UI: Clarity and Color on the Presentation Layer. Feb 26 2009 [Электронный ресурс]. – Режим доступа: <https://community.devexpress.com/blogs/markmiller/archive/2009/02/26/great-ui-clarity-and-color-on-the-presentation-layer.aspx>, своб.
3. Пыхтин С.В. Таблица безопасных цветов для разработки дизайна сайта [Электронный ресурс]. – Режим доступа: <http://www.antula.ru/palitra.htm>, своб.
4. Чуракова А. Как проектировать для людей с цветовой слепотой [Электронный ресурс]. – Режим доступа: <https://infogra.ru/ui/dizajn-dlya-daltonikov-5-faktorov-uspeha>, своб.
5. Рассматриваем все преимущества минимализма в веб-дизайне [Электронный ресурс]. – Режим доступа: <https://webformula.pro/article/rassmatrivaem-vse-preimushchestva-minimalizma-v-veb-dizayne/>, своб.



Варганов Дмитрий Анатольевич

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107с

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: varganovda@gmail.com

УДК 004.4'22

ОБЗОР СРЕДСТВ ГЕНЕРАЦИИ ПРОГРАММНОГО КОДА НА ОСНОВЕ UML

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 продукты для автоматизированного проектирования программного обеспечения (CASE-средства), произведено описание инструментов, осуществляющих генерацию программного кода на основе моделей унифицированного языка моделирования (UML), их возможностей.

Ключевые слова: CASE-средства, UML, автоматизированная генерация программного кода.

Жизненный цикл веб-сайтов, как и другого программного обеспечения (ПО) основывается на одной из наиболее распространенных методологий разработки «Унифицированный процесс разработки программного обеспечения» («The Unified Software Development Process») [1]. Он состоит из следующих этапов:

1. появление идеи о создании программного продукта;
2. сбор требований;
3. составление технического задания;
4. проектирование разработки;
5. написание программного кода;
6. тестирование и отладка;
7. внедрение;
8. поддержка.

Наибольшие трудовые временные затраты приходятся на этапы проектирования и написания программного кода, в связи с чем в области разработки ПО появляются новые и усовершенствованные методики и технологии, способствующие многократному снижению издержек. Одной из таких технологий является генерация кода программы на основе концептуальных моделей. Это становится возможным благодаря CASE-средствам (Computer-Aided Software Engineering, автоматизированное проектирование программного обеспечения), обеспечивающим специалистов большим количеством инструментов, начиная с анализа требований и заканчивая конфигурационным управлением на стадии внедрения [2]. Далее в данной работе рассмотрены CASE-продукты, осуществляющие генерацию программного кода, на основании графического языка UML (Unified Modelling Language, унифицированный язык моделирования).

В процессе разработки ПО осуществляется создание сопроводительной технической документации. «Техническая документация – набор документов, используемых при проектировании (конструировании), создании (изготовлении) и использовании (эксплуатации) каких-либо технических объектов: зданий, сооружений, промышленных товаров, программного и аппаратного обеспечения». Согласно ГОСТу [3, 4], в нее принято включать описание, содержащее сведения о логической структуре программы, а также пояснительную записку, определяющую алгоритм работы продукта, особенности эксплуатации и технико-экономических решений, принятых в процессе разработки [4]. Помимо этого, в пакет документов могут входить модели языка UML, предназначенного для фиксации артефактов, визуализации и проектирования структуры. Использование унифицированного графического языка в процессе разработки продукции имеет следующие дополнительные преимущества:

1. возможность взаимодействия с людьми, не имеющими технического образования и знаний в области разработки ПО;
2. планирование временных затрат;
3. ввод правил и ограничений в проекте;
4. рефакторинг кода;
5. кодогенерация.

Последнее стало возможным за счет появления новой архитектуры MDA (Model Driven Architecture, архитектура, управляемая моделями) в начале 2000 годов, идея которой заключается в предельной автоматизации генерации кода. Позднее появляется еще один подход, называемый MDD (Model Driven Development, разработка, управляемая моделями). Этот подход отличается от предыдущего в необходимости предварительной трансформации моделей.

На основе описанных нововведений многие компании произвели выпуск CASE-средств для обеспечения команд-разработчиков всем возможным, начиная с анализа

требований и заканчивая процессом конфигурационного управления. Рассмотрим возможности некоторых самых распространенных продуктов.

Rational Rose – программный продукт компании IBM, на момент написания работы является одним из лидеров рынка аналогичных средств. Основные предназначения программы заключаются в автоматизации этапов анализа, проектирования, разработки (генерация программного кода для различных языков программирования, таких как: Ada, ANSI C++, C++, CORBA, Java, Java EE, Visual C++ и Visual Basic на основе визуальных моделей UML) и выпуска проектной документации. Продукт компании IBM использует объектно-ориентированную методологию и имеет средства, позволяющие повторно применять программные компоненты в новых проектах [5].

ARIS Designer, ARIS Architect – два программных продукта компании Software AG, работающие в симбиозе. Имеют возможности, аналогичные продукту компании IBM. Однако генерация кода производится для языков семейств C/C++, Java, .NET и Visual Basic. Из отличий также можно выделить наличие глобального репозитория и дополнения в виде уникального веб-браузера, позволяющего многим пользователям работать одновременно. Помимо описанных достоинств, имеют обособленные механизмы защиты и непротиворечивой передачи данных. Основаны на объектно-ориентированном подходе [6].

Component Modeler/Paradigm Plus – программный продукт, разработанный компанией CA, распространявшийся в рамках пакета AllFusion Modelling Suite, в настоящее время развитие проекта остановлено. Средство предназначено для проектирования, визуализации и поддержки программных продуктов. Предоставляет возможности прямого и обратного проектирования, т.е. позволяет помимо генерации программного кода на основе моделей UML производить реверсивную генерацию диаграмм, вне зависимости от изменений. Также имеет встроенный генератор документации и отчетов. Используемый подход генерации кода неизвестен, совместим с языками Visual C++, C, Java, Visual Basic и CORBA.

Microsoft Visio – программный продукт компании Microsoft. Его предназначение заключается в универсальности и максимальной совместимости, позволяет создавать временные шкалы, карты процессов, блок-схемы, схемы организационной структуры, архитектуры IT-систем, схемы этажей и т.д. Поддерживает обновление данных в текущий момент времени. Возможность генерации программного кода отсутствует.

Oracle JDeveloper – программный продукт компании Oracle. Представляет собой интегрированную среду разработки программных продуктов. Обеспечивает возможности, начиная от проектирования архитектуры будущего приложения и заканчивая рефакторингом программного кода, включая создание диаграмм на основе языка UML. Допускает работу со следующими языками: Java (основная ориентация), XML, SQL, PL/SQL, HTML, JavaScript, CSS, BPEL и PHP. Возможность генерации программного кода отсутствует.

Проведенный анализ показал, что в настоящее время на рынке продуктов по проектированию и разработке программ представлено большое количество инструментов, позволяющих снизить затраты ресурсов на создание ПО на различных этапах жизненного цикла. Следовательно, область остается актуальной в настоящее время.

Литература

1. Jacobson I., Booch G, Rumbaugh J. The Unified Software Development Process. – Addison-Wesley, 1999. – 498 p.
2. Бориско С.Н., Васильев И.Н., Лобейко В.И. Оптимизация выбора объектно-ориентированного CASE-средства в жизненном цикле программного продукта // Изв. Волгоградского государственного технического университета. – 2011. –

№ 10(3). – С. 42–45.

- ГОСТ 19.101-77. Виды программ и программных документов. – Введен 01.01.1980. – М.: Изд-во стандартов, 1980. – 4 с.
- Документация по ГОСТу: зачем это современным веб-разработчикам [Электронный ресурс]. – Режим доступа: <http://popel-studio.com/blog/article/dokumentacija-po-gostu.html>, своб.
- Гхош К.П., Бутенко Л.Н. Сравнительный анализ CASE средств // Изв. Волгоградского государственного технического университета. – 2009. – № 7(12). – С. 35–39.
- CASE-средства: в борьбе со сложностью мира [Электронный ресурс]. – Режим доступа: <http://www.interface.ru/home.asp?artId=986>, своб.



Верхоумов Дмитрий Олегович

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107с

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: verkhovmov@yandex.ru

УДК 004.6

**ОБЗОР ТЕХНОЛОГИЙ ОБМЕНА ДАННЫМИ В ВЕБ-ПРИЛОЖЕНИИ
В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ**

Верхоумов Д.О.

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 рассмотрены основные технологии обмена данными между веб-приложением и клиентом в режиме реального времени. Представлено описание работы каждого подхода. Выделены основные недостатки и приведены рекомендации по дальнейшему развитию технологий обмена данными в реальном времени.

Ключевые слова: веб-приложение, обмен данными, AJAX, Polling, Long polling, Server-Sent Events, WebSocket, WebRTC.

Многие современные веб-приложения используют технологии обмена данными между клиентом и сервером в режиме реального времени. Это обусловлено необходимостью своевременной передачи информации пользователю и мгновенного реагирования на взаимодействия пользователя с интерфейсом. Подобные технологии позволяют создавать веб-приложения любой сложности, например, сервисы для мгновенного обмена сообщениями, социальные сети, сервисы для голосовых и видеоконференций, многопользовательские браузерные онлайн-игры, сервисы мониторинга в режиме реального времени и многое другое. В зависимости от решаемых задач и требований к веб-приложению могут быть использованы следующие технологии: AJAX, Polling, Long polling, Server-Sent Events, WebSocket, WebRTC. Каждая из перечисленных технологий имеет преимущества и недостатки, а также отличается схемой работы.

Стандартный протокол HTTP, реализующий передачу данных по схеме «запрос-ответ», не подходит для работы в режиме реального времени, так как не может

отправлять данные клиенту принудительно и в одностороннем порядке. Помимо этого, за счет информации, передаваемой в заголовке запроса, общий объем передаваемых данных, умноженный на количество запросов, избыточен. В результате происходит генерация лишнего трафика, что, в свою очередь, сказывается на отзывчивости самого веб-приложения, когда на выполнение одного запроса требуется большое количество времени.

Одной из самых простых технологий для организации схемы «запрос-ответ» между клиентом и сервером в режиме реального времени является AJAX. Данная технология использует стандартный HTTP-протокол. Для создания запроса используется специальный программный код, выполняющийся на стороне клиента при совершении события, например, при нажатии на кнопку. Сервер, в свою очередь, в качестве ответа возвращает только необходимые данные в формате XML или JSON, благодаря чему происходит значительное сокращение трафика [1]. Обработка запросов происходит в фоновом режиме, поэтому пользователь может параллельно совершать другие действия.

С целью улучшения стандартного подхода при использовании AJAX и расширения его функциональных возможностей был создан новый подход опросов Polling. Под опросом понимается периодическая отправка запросов от клиента к серверу и получение ответов. В рамках данного подхода специальный программный код на стороне клиента с заданной периодичностью отправляет запрос на сервер [2]. Сервер, в свою очередь, в качестве ответа либо возвращает актуальную информацию, либо сообщает о ее отсутствии. К основным недостаткам такого подхода можно отнести следующие:

- выполнение лишних запросов, когда актуальная информация на сервере отсутствует;
- частота поступления актуальной информации обусловлена частотой опроса и временем выполнения запроса;
- избыточный объем передаваемой информации, обусловленный передачей полного пакета HTTP-заголовков при каждом новом запросе.

Для решения проблемы избыточного количества запросов был создан подход, называемый длительным опросом Long polling, который также основан на технологии AJAX. Отличительной чертой данного подхода является то, что пока на сервере не появятся актуальные данные, он не вернет ответ на запрос [2]. После того как клиент получит ответ, будет сформирован и отправлен новый запрос для получения более свежей информации.

Ранее описанные технологии обмена данными подразумевают возможность реализации без внесения особых изменений в серверную часть. Достаточно лишь создать отдельную страницу, которая будет принимать стандартный HTTP-запрос через AJAX. Технологии, о которых пойдет речь далее, используют специальные протоколы передачи данных, не имеющие ничего общего с обычным протоколом HTTP.

Одной из таких технологий является Server-Sent Events (SSE). SSE стандартизируется организацией W3C как часть спецификации HTML5, благодаря чему решается проблема совместимости с различными браузерами. Современный стандарт SSE позволяет браузеру создавать специальный объект EventSource, который сам обеспечивает соединение с сервером, пересоздает соединение в случае обрыва и генерирует события при поступлении данных. При создании нового соединения на сервер отправляется полный пакет HTTP-заголовков. После подключения сервер в любой момент времени может отправлять данные клиенту, например, при появлении актуальной информации. Недостатком данного подхода является использование односторонней передачи данных «сервер-клиент», который в определенных условиях может быть преимуществом. Также в SSE полностью отсутствует поддержка браузера Internet Explorer (IE).

Полноценной альтернативой SSE является технология WebSocket. Это протокол, реализующий двунаправленную полнодуплексную связь, работа которой обеспечивается поверх протокола TCP [3]. WebSocket стандартизирован по RFC 6455 и является частью спецификации HTML5, поэтому совместим с последними версиями многих известных на сегодняшний день браузеров [4]. WebSocket более совершенен, так как в нем решены проблемы описанных ранее технологий. После создания соединения по данному протоколу клиент и сервер могут одновременно передавать и получать актуальную информацию, а за счет особого формата пакетов данных, объем дополнительно передаваемой служебной информации сведен к минимуму. В качестве еще одной особенности WebSocket можно указать поддержку зашифрованного соединения, которое реализуется над протоколом HTTPS. Основным недостатком данной технологии является отсутствие поддержки устаревшими браузерами.

В отличие от SSE и WebSocket, инновационная технология WebRTC использует сервер только для представления двух и более клиентов друг другу, чтобы поддерживать между ними соединение без участия сервера. Данная технология является веб-коммуникацией в реальном времени, может использовать протоколы UDP, TCP или более абстрактные слои, и предназначена для организации передачи потоковых данных между браузерами или другими приложениями, поддерживающими данную технологию. Стоит отметить, что WebRTC стандартизируется организацией W3C и является расширением HTML5. На данный момент технология находится в разработке и не имеет официальной версии, однако, это не мешает разработчикам создавать пробные веб-приложения для аудио- и видеоконференций.

Основываясь на представленных данных можно отметить незаменимость каждой из описанных технологий. Вопреки тому, что некоторые из них являются преимуществом, существуют задачи, решение которых возможно только при определенной схеме работы, поэтому, например, Long polling не сможет заменить стандартный AJAX, и наоборот. Главным и в то же время общим недостатком всех рассмотренных технологий является слабая или полностью отсутствующая поддержка устаревших браузеров. Частично данную проблему можно решить с помощью библиотек, использующих сразу несколько технологий для обмена данными в режиме реального времени, например, Socket.IO для Node.js [5].

Наиболее перспективным является развитие технологии передачи потоковых данных WebRTC. Во-первых, это приведет к возможности создания распределенной архитектуры не только на уровне серверного оборудования, но и на клиентском уровне. В итоге пользователи смогут напрямую обмениваться не только потоковыми, но и любыми другими данными, включая гипертекстовую разметку. Это в значительной мере снизит нагрузку на оборудование, увеличит отзывчивость веб-приложения и, как следствие, уменьшит материальные затраты на техническое сопровождение. Во-вторых, это позволит обмениваться конфиденциальными данными между двумя и более клиентами без участия сервера с помощью «end-to-end» шифрования.

Литература

1. Пауэлл Т. Ajax. Настольная книга программиста. – М.: Эксмо, 2009. – 712 с.
2. In what situations would AJAX long/short polling be preferred over HTML5 WebSockets? // Stack Overflow [Электронный ресурс]. – Режим доступа: <http://stackoverflow.com/questions/10028770/in-what-situations-would-ajax-long-short-polling-be-preferred-over-html5-websock>, своб.
3. WebSocket // Современный учебник Javascript [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/websockets>, своб.
4. RFC 6455 – The WebSocket Protocol // Internet Engineering Task Force (IETF) [Электронный ресурс]. – Режим доступа: <https://tools.ietf.org/html/rfc6455>, своб.
5. Steven Holzner Socket.IO Real-time Web Application Development. – 2008. – P. 475.

**Волобуев Никита Геннадиевич**

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4108Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: chester.19948@gmail.com

**Готская Ирина Борисовна**Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна,

д.педагог.н., профессор

e-mail: iringot@mail.ru

УДК 004.4

СРАВНЕНИЕ WEB-ФРЕЙМВОРКОВ PYTHON**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, написанных с помощью скриптового языка программирования Python. Выявлены основные достоинства и недостатки для: Python, Django и Flask.

Ключевые слова: Python, web-фреймворк, Django, Flask, web-разработка, web-приложения.

В настоящее время с развитием сетевых технологий глобальная сеть Интернет стала использоваться во многих областях нашей жизни. Все больше и больше людей пользуются интернетом – посещают разнообразные сайты и пользуются приложениями в смартфонах. В связи с этим появилось множество технологий, которые предоставляют возможность создания и разработки сайтов и web-приложений. Учитывая актуальность данной темы, в настоящей работе ставилась следующая **цель** – сравнить самые популярные web-фреймворки, написанные с помощью скриптового языка программирования Python. Python – универсальный язык программирования, применяемый, в том числе и в web. С технической точки зрения web-приложение на Python – это полноценное приложение, загруженное в память, обладающее своим внутренним состоянием, сохраняемым от запроса к запросу. На декабрь 2016 года Python занимает 4 место в рейтинге TIOBE programming community index. Данный рейтинг отражает популярность языков программирования, опираясь на подсчеты результатов поисковых запросов, содержащих название языка. Сам по себе язык обладает интуитивно понятным синтаксисом и располагает огромным количеством библиотек, предназначенных для решения совершенно не похожих друг на друга задач. Язык обладает целым рядом преимуществ:

- открытая разработка;
- довольно прост в изучении, особенно на начальном этапе;
- особенности синтаксиса стимулируют программиста писать хорошо читаемый код;
- предоставляет средства быстрого прототипирования и динамической семантики;
- имеет большое сообщество, которое взаимодействует и поддерживает начинающих программистов;

- множество полезных библиотек и расширений языка можно легко использовать в своих проектах благодаря предельно унифицированному механизму импорта и программным интерфейсам;
- механизмы модульности хорошо продуманы и могут быть легко использованы;
- в Python каждое число, строка, функция, модуль и т.п. является объектом в контексте объектно-ориентированного программирования (ООП), но при этом объектный подход не навязывается программисту.

Однако у данного языка также присутствуют и недостатки:

- не вполне удачная поддержка многопоточности;
- на Python создано не так много качественных программных проектов по сравнению с другими универсальными языками программирования, например, с Java;
- отсутствие коммерческой поддержки средств разработки (хотя эта ситуация со временем меняется);
- изначальная ограниченность средств для работы с базами данных;
- тесты показывают меньшую производительность Python по сравнению с основными Java VM, что создает этому языку репутацию медленного [1, С. 183].

Наиболее распространенным из всех доступных web-фреймворков Python на данный момент является Django – согласно статистике на декабрь 2016 года самый популярный среди сообщества программистов web-ресурс Stack Overflow содержит 131,987 тысяч вопросов от разработчиков со всего мира. К тому же данным фреймворком пользуются такие гиганты как Google, Dropbox, Washington Times, Yandex и многие другие. Второе место по популярности среди разработчиков занимает Flask, ему посвящено 37,894 тысяч вопросов, касающихся разработки.

Flask является микрофреймворком языка Python, в первую очередь он предназначен для небольших проектов с простыми требованиями. Приставка «микро» не значит, что web-приложение может полностью поместиться в одном файле, написанном на Python. Таким же образом это не означает, что у фреймворка Flask ограниченный функционал. «Микро» означает, что Flask ставит своей задачей придерживаться простого, но расширяемого ядра. Flask не сможет решать за разработчика многие вещи, например, какую базу данных выбрать для проекта, но решения, которые он может применять, можно легко изменить под свои потребности, например, какой из движков для работы с шаблонами использовать.

С целью определения проектов, для которых лучше подходит Django или Flask, сравним основные компоненты этих фреймворков (структуру проекта; минимальные приложения; модульную систему проектов; встроенный интерфейс администрирования; документацию фреймворков и их модулей).

Flask располагает огромным количеством различных способов организовывать код и структуру проекта. При создании нового проекта из-за отсутствия строгой структуры не всегда понятно, как лучше организовать файлы фреймворка, возможна ситуация, что средств, предустановленных во Flask, будет недостаточно для создания полноценного приложения, и придется искать дополнительные модули. У неопытного программиста это может вызвать проблемы, так как придется разбираться не только с фреймворком, но и с дополнительными модулями, которые в большинстве случаев имеют не очень хорошо структурированную документацию.

Для больших проектов Flask использует концепцию blueprints для создания элементов приложения и обеспечения поддержки общих шаблонов непосредственно в самом приложении или между приложениями.

В свою очередь, структура проектов на фреймворке Django в большинстве случаев имеет практически идентичный вид, благодаря этому в будущем не должно возникнуть никаких проблем при переходе с одного проекта на другой.

Документация обоих фреймворков очень высокого уровня, но из-за того, что Django позиционирует себя как «все в одном», большая часть модулей данного фреймворка имеет также

отличную документацию и высокую поддержку сообщества. Модули, предназначенные для Flask, к сожалению, не обладают столь хорошей документацией [2, 3].

Подводя итоги вышесказанного, можно сделать следующие выводы. Python – довольно популярный язык программирования, обладающий большим количеством разнообразных web-фреймворков, каждый из которых имеет свои плюсы и минусы. Самым популярным фреймворком языка Python является Django. Он характеризуется широкой аудиторией пользователей и активной командой разработчиков, которые каждый год выпускают новую версию всем известного фреймворка. В целом вся структура веб-приложения, написанного с помощью Django, строго определена. Методы хранятся только в одном файле и нигде больше, модели располагаются в другом, таким образом, абсолютно все компоненты имеют свое заранее подготовленное место в структуре фреймворка. При разработке на нем не требуется думать об архитектуре проекта, все заранее определено создателями фреймворка.

Существует и другой, не менее популярный микрофреймворк, широко используемый среди web-разработчиков, пишущих на языке Python. Flask позиционирует себя как расширяемый фреймворк. Это означает наличие только самого необходимого минимума, достаточного для его работоспособности, но в то же время имеется возможность посредством большого количества расширений придать проекту необходимый уровень. Все вышесказанное позволяет предположить, что выбор фреймворка зависит как от сроков, выделенных на создание нового приложения, так и от субъективного опыта разработчика.

Литература

1. Одинамамадов Ф.И. Web-программирование // Сб. научных трудов II Международной конференции. – 2015. – С. 180–185.
2. Мелихов А.Ю., Юхимук Р.Ф. Создание web-приложения для генерации отчетов на базе каркасной системы управления контентом «Django» // Вестник Югорского государственного университета. – 2011. – № 3(22). – С. 27–30.
3. Васильев П.А. Web-программирование на языке Python. Фреймворки Django, Flask // Наука, техника и образование. – 2016. – С. 38–39.



Волошин Олег Владимирович

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4106

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: o.voloshin@protonmail.com

УДК 004.422.83

МЕТОДИКИ СТРУКТУРИЗАЦИИ ЭВРИСТИЧЕСКИХ МЕТОДОВ В ПРОИЗВОДСТВЕННЫХ ПРОЦЕССАХ

Волошин О.В.

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

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

Проведен анализ основных способов формализации эвристических методов на основе практик, используемых в производстве гибкой продукции и в эко-дизайне. Предложена собственная

система структуризации эвристических методов для начала проектирования системы управления эвристических методов и их визуализации.

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

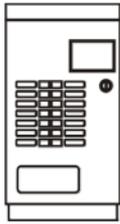
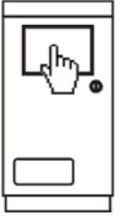
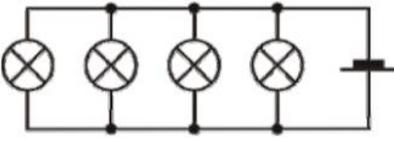
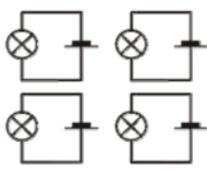
Эвристические методы, принципы или руководства к проектированию являются инструкциями, помогающими ориентироваться в принятии решения по отношению к конкретной производственной задаче. Они являются простым инструментом, который способствует интеграции предложенных требований в рамках конструкции планируемой продукции. Однако в повседневной работе удобство использования данного инструмента является невысоким с точки зрения методик проектирования. В частности, текущие решения имеют следующие недостатки:

- отсутствие должного количества контекстной информации, оказывающей помощь в ситуационном выборе необходимых методов;
- необходимость адаптации степени типичности/специфичности методов для конкретной ситуации;
- неочевидность взаимосвязей и необходимых компромиссов, возникающих в рамках масштабных систем рекомендаций;
- сложный процесс ориентирования, модификации или добавления новых информационных блоков в базы данных по мере поступления новых решений.

Целью работы являлась разработка концепции и прототипа хранилища и утилиты для визуализации, которые помогут решить вышеперечисленные проблемы.

Чаще всего эвристические методы представлены в виде списка или таблицы кратких высказываний (табл. 1), которые в процессе разработки интерпретируются проектировщиками.

Таблица 1. Пример конструкторских эвристических методов [1]

№	Эвристический метод	Негибкая продукция	Гибкая продукция
1	Программные решения вместо аппаратных		
2	Использование автономных модулей		

Использование эвристических методов получило широкое распространение за счет отсутствия высоких трудовых и материальных затрат как обязательного требования для создания эвристических систем, что потенциально делает эвристические методы мощным, но в то же время простым инструментом поддержки принятия промышленных решений. Тем не менее, наряду с развитием отрасли производственного конструирования, закономерно ожидать определенной степени формализации и документирования даже элементарных инструментов поддержки.

Другой возникающей при использовании эвристических методов проблемой является недостаток контекстной информации. В связи с этим у читателя могут возникать вопросы, связанные с определением перечня необходимых или рекомендуемых объектов для применения конкретной инструкции. Впервые контекстная информация как атрибут эвристических методов была предложена Кливлендом Гиббоном (табл. 2).

Таблица 2. Пример списка атрибутов эвристических методов [2]

Название	Описание
Имя	Описание эвристического метода
Цель	Цель эвристического метода
Обоснование	Краткое описание проблемы
Последствия	Результаты применения эвристического метода
Контекстная информация	Важные конструкторские особенности, которые влияют на поведение эвристического метода
Эвристические взаимосвязи	Методы, имеющие прямую взаимосвязь с данным
Предложения	Перечень рекомендуемых действий на случай нарушения выполнения эвристического метода

На данном примере видно, что предложенный атрибут раскрывает технические детали, связанные с конструкцией объектов применения эвристической инструкции. Однако существуют и другие трактовки – так, например, Джулиан Сарнес [3] предлагает более эффективное применение данному атрибуту. Он описывает контекстную информацию как «определение системы продукции, в которой данный эвристический метод будет», т.е. в его определении контекстной информацией является область применения эвристических инструкций. В отдельных случаях эвристические методы могут быть релевантными для всех продуктов, только тогда необходимость в таком атрибуте отпадает.

Кроме того, контекстная информация может содержать не только область применения, но и другую вспомогательную информацию (например, ответственная за выполнение метода единица или степень внедрения в систему). Таким образом, можно группировать контексты по атрибутам так, что при желании читатель смог получить всю необходимую информацию, релевантную конкретной инструкции.

Возвращаясь к вопросу отсутствия структуризации эвристических систем, положительным фактором может являться их представление в виде пошаговых инструкций или же систем инструкций, подкрепленных взаимосвязями. Повторно обратимся к примеру эвристической структуры, представленной в виде направленной диаграммы (рисунок). Рисунок графически представляет совокупность эвристических методов в виде структурированной системы звеньев и взаимосвязей между ними. На верхнем уровне метод «Уменьшение стоимости жизненного цикла» является типичной эвристической инструкцией и формулирует общие направления принятия дальнейших решений в процессе проектирования.



Рисунок. Фрагмент примера структурированной эвристической системы, представленной в виде направленной диаграммы [4]

Обращаясь к дочерним элементам системы (например, «Снижение затрат на производство») читатель раскрывает один из путей реализации инструкции в корне структуры. Двигаясь дальше по диаграмме к дочерним звеньям, инструкции становятся все более и более специфичными и предоставляют более детальные руководства к действию. Такие эвристические системы являются более наглядными и простыми для понимания читателем, позволяя пошагово планировать и проследить процесс внесения изменений.

Итак, среди итоговых критериев можно выделить следующие:

- использование четко сформулированных, понятных и кратких предложений непосредственно в тексте инструкции эвристического метода;

- составлять для одной идеи не более одного эвристического метода, ровно также стоит отказаться от использования нескольких идей для создания одной инструкции;
- избегать составления противоречащих друг другу эвристических методов;
- составлять эвристические методы при условии должного обоснования для их применения;
- учитывать при составлении эвристических систем возможный разброс в уровне профессиональной подготовки, опытности и принадлежности к разным сферам деятельности читателя;
- при необходимости использовать дополнительную контекстную информацию;
- дополнять текстовую составляющую эвристических методов визуальной поддержкой, содержащей необходимые технические детали (например, необходимые наименования компонентов продукции, их взаимодействие и т.д.);
- балансировать наличие визуальной составляющей с содержанием эвристической инструкции;
- допускать незначительный уровень избыточности эвристических систем для обеспечения их стабильности;
- группировать эвристические методы соответственно их определенному контексту.

Данные критерии, в рамках продолжения данной работы, будут являться основой для проектирования модели данных для утилиты структуризации эвристических методов [5].

Литература

1. Fu K.K., Yang M.C. and Wood K.L. Design Principles: The Foundation of Design // International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. – 2015. – 10 p.
2. Bischof A., Blessing L. Guidelines for the development of flexible products // Proceedings of the 10th International Design Conference. – 2008. – P. 289–300.
3. Sarnes J., Kloberdanz H. Heuristic guidelines in ecodesign // Proceedings of the 20th International Conference on Engineering Design (ICED 15). – 2015. – V. 1. – 15 p.
4. Bonvoisin J., Mathieux F., Domingo L. & Brissaud D. Design for energy efficiency: proposition of a guidelines-based tool // In International Design Conference. – 2010. – P. 629–638.
5. Bischof A. Developing flexible products for changing environments: diss. PhD thesis. – Technische Universitaet Berlin, 2010. – 233 p.



Глебов Родион Геннадьевич

Год рождения: 1981

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра аппаратно-программных комплексов и вычислительной
техники, группа № S4100

Направление подготовки: 09.04.01 – Информатика и вычислительная
техника

e-mail: thomas_cp92@mail.ru

УДК 004.453

ИССЛЕДОВАНИЕ ПРОЦЕССОВ ЗАЩИЩЕННОСТИ В КОРПОРАТИВНЫХ СЕТЯХ НА ОСНОВЕ АППАРАТНО-ПРОГРАММНОГО КОМПЛЕКСА VipNet

Глебов Р.Г.

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

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе проведено сравнительное исследование процессов защищенности в корпоративных сетях на основе аппаратно-программного комплекса ViPNet – построенного на основе VPN-соединений, проведено сравнение данного комплекса с обычными соединениями на основе VPN, выделены основные преимущества комплекса ViPNet для защиты информации в корпоративных сетях.

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

В работе рассмотрены эффективные средства создания защищенной и отказоустойчивой корпоративной компьютерной сети на основе аппаратно-программного комплекса ViPNet. Приведены основные ее достоинства.

Исследуемая тема рассматривалась в работах некоторых авторов. Стоит отметить издания В.В. Гусева [1, 2] «Администрирование системы защиты информации ViPNet», «Программно-аппаратные комплексы ViPNet», «Официальный курс по организации виртуальных защищенных сетей ViPNet», в которых дано исчерпывающее представление о технологии ViPNet и рассмотрены основы администрирования программно-аппаратных комплексов ViPNet.

Аппаратно-программный комплекс ViPNet базируется на технологиях VPN. Аббревиатура VPN расшифровывается не только как Virtual Private Network (виртуальная частная сеть), но и как Virtual Protected Network, т.е. виртуальная защищенная сеть.

Термин «private» имеет два основных значения: частный (собственный) и конфиденциальный (закрытый). VPN – это частная сеть передачи данных, использующая открытую телекоммуникационную инфраструктуру и сохраняющая при этом конфиденциальность передаваемых данных, посредством применения протоколов туннелирования и средств защиты информации. Задача VPN-технологий состоит в максимальной степени отделения потока данных одного предприятия от потока данных всех других пользователей сети общего пользования. При этом должна быть конфиденциальность данных и их защита от несанкционированного доступа при гарантированных качествах обслуживания в сетях. Однако обычные технологии VPN не позволяют широко известными антивирусными программными средствами производить полную проверку трафика без потери скорости доступа в сети. В связи с этим для защиты в корпоративной сети используются разнообразные сочетания средств обеспечения безопасности. В рамках работы описаны основные преимущества комплекса ViPNet [3].

Комплекс состоит из трех основных компонентов:

1. координатор – это аппаратно-программный комплекс, состоящий из специального оборудования с поддержкой: сервер-маршрутизации (обеспечивает маршрутизацию пакетов между участниками обмена); сервер-IP пакетов (обеспечивает регистрацию и предоставление информации о текущих IP-адресах и способах подключения объектов сети); сервер ViPNet-Firewall (обеспечивает работу защищенных компьютеров сети в VPN от имени одного адреса и работу защищенных компьютеров локальной сети через другие Firewall или устройства с поддержкой технологии NAT, а также туннелирование пакетов в защищенное соединение от заданных адресов незащищенных компьютеров); ViPNet-сервер открытого Интернета (обеспечивает организацию безопасного подключения части компьютеров локальной сети к Интернету без их физического отключения от локальной сети организации);
2. администратор – это совокупность двух разных и независимых программ: центр управления сетью (обеспечивает централизованное управление сетью и формирует структуру VPN совместно с формированием и управлением «логикой» работы VPN; обеспечивает централизованное обновление программного обеспечения и функционала компонентов VPN; мониторинг событий VPN; удаленное управление ресурсами VPN) и удостоверяющий ключевой центр (обеспечивает участников сети VPN ключевой информацией и выполняет функции удостоверяющего центра);
3. клиент – программный модуль, позволяющий без дополнительных знаний и настроек встроенного и дополнительного функционала операционной системы защитить компьютер от

попыток несанкционированного доступа, как из глобальной, так и из локальной сети, а также позволяет управлять терминалом, где установлен модуль, а также осуществлять фильтрацию трафика по уровню сервиса и протокола). С его помощью происходит контроль активности приложений и блокировка нежелательных приложений без участия пользователя.

Все эти возможности модуля реализуются благодаря основному программному модулю ViPNet-драйвер. Драйвер работает на уровне IP-пакетов до того, как они передаются стандартному стеку TCP/IP, поэтому он контролирует весь IP-трафик, поступающий в сеть и из сети, и обеспечивает защиту для любых приложений, не меняя их привычного использования пользователями [3]. Это позволяет внедрять ViPNet как систему информационной безопасности без «насилия» над принятыми в корпорации бизнес-процессами и приложениями, поддерживающими эти бизнес-процессы, начиная от простейших Web-приложений и кончая сложнейшими ERP-системами. Основными функциями ViPNet-драйвера является принятие решения по каждому отправляемому или получаемому IP-пакету. Для каждого отправляемого IP-пакета осуществляется одно из трех действий: пропустить в незащищенном виде, не пропускать вообще, зашифровать и отправить в зашифрованном виде. Драйвер сетевой защиты ViPNet для всех типов исходящих или туннелируемых пакетов, IP-адреса которых сопоставлены для определенного идентификатора, производит упаковку (инкапсуляцию) этих пакетов в другой пакет протокола UDP или IP/241. Протокол UDP-драйвер автоматически использует, если считает, что на пути следования IP-пакета присутствуют устройства, осуществляющие преобразование адресов (NAT). В ином случае используется более экономичный с точки зрения добавляемой избыточной информации протокол IP/241, в котором отсутствует UDP-заголовок [4].

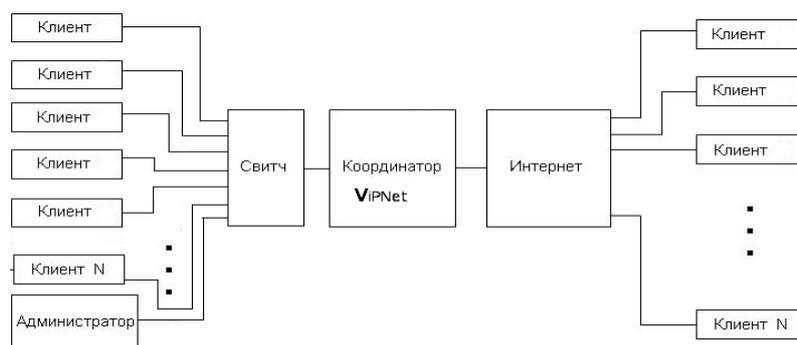


Рисунок. Структурная схема защищенной сети

На основе всего сказанного можно построить структурную схему защищаемой сети. Она представлена на рисунке.

Основными преимуществами комплекса ViPNet перед другими методами доступа по технологии VPN можно отметить: высокая производительность шифрования и фильтрации трафика в реальном времени позволяет осуществлять защиту трафика служб голосовой и видеосвязи в сетях TCP/IP, а также обеспечивать одновременную работу с ресурсами разных сегментов корпоративной сети. Также защита канала не влияет на работу сторонних приложений на компьютере пользователя.

Литература

1. Гусев В.В. Программно-аппаратные комплексы ViPNet. – М.: Горячая линия–Телеком, 2014. – 354 с.
2. Гусев В.В. Администрирование системы защиты информации ViPNet. – М.: Горячая линия–Телеком, 2013. – 379 с.
3. Чефранова А.О. Технология построения VPN ViPNet. – М. Горячая линия–Телеком, 2009. – 289 с.
4. Кабакова Н.В., Чефранова А.О., Алабина Ю.Ф. Система защиты информации ViPNet. Курс лекций. – М.: ДМК-Пресс, 2015. – 456 с.



Головатая Елена Николаевна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

УДК 004.92

АНАЛИЗ МЕТОДОВ БРЕНДИНГА ИНТЕРНЕТ-СТАРТАПОВ

Головатая Е.Н.

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

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

Работа посвящена анализу методов брендинга интернет-стартапов. Рассмотрены понятия бренда и брендинга. Произведено сравнение создания и продвижения брендов интернет-стартапа и офлайн-компаний. В работе описаны основные методы создания бренда и инструменты его продвижения.

Ключевые слова: бренд, брендинг, создание бренда, продвижение бренда, стартап, интернет-проект, интернет-стартап.

Термином «бренд» обозначается неосязаемый набор характеристик продукта, его названия, упаковки, цены, истории, репутации и способа продвижения в сочетании с впечатлением, которое он производит на потребителя, и обещанными ему автором бренда определенными преимуществами.

Под «брендингом» понимается процесс создания и управления брендом, включающий ряд действий, основной целью которых является формирование у потребителя определенного образа рекламируемой торговой марки, компании, товара или услуги [1].

Можно выделить две основные составляющие бренда, вместе образующие единое целое.

1. Собственно, бренд, то, чем он является. Это сама торговая марка, средства внешней визуальной идентификации, позиционирование, уникальность, миссия бренда, понимание места на рынке. Это самое главное, фундамент любого бренда. По мнению гуру маркетинга Джека Траута, именно эта составляющая многих торговых марок нуждается сейчас в улучшении. С его мнением трудно не согласиться: создать хорошую основу трудно, а цена ошибки велика, так как сколько бы компания ни продвигала слабо продуманный, «недоработанный» бренд, он не принесет желаемой эффективности.
2. Представления о бренде, то, что о нем думают. Это узнаваемость марки на рынке, ассоциации с ней, лояльность потребителей и приверженность к бренду. По мнению автора, это не менее важная часть, поскольку даже идеально разработанный бренд нуждается в продвижении [2].

Сейчас, к сожалению, многие компании не используют все возможности брендинга, они бездумно идут по стандартному пути: создают фирменный стиль и, в лучшем случае, придумывают позиционирование, после чего начинают тотальную рекламную кампанию во всех каналах. Конечно, достичь высокой эффективности в таком случае не получится.

Следовательно, при соблюдении основных правил брендинга можно получать хорошую прибыль, и для этого совсем не обязательно иметь многомиллионный маркетинговый бюджет.

Независимо от размеров бренда ключ к успеху – сделать так, чтобы потребители рассказывали друг другу о компании или продукте. Благодаря этому будет формироваться узнаваемость бренда, постепенно сформируется лояльность и, в конечном итоге, приверженность.

Давно известно, что наиболее эффективная реклама – это рекомендация знакомого человека, поэтому компании сейчас повсеместно пытаются заставить потребителей говорить о бренде.

Брендинг в интернет-стартапах имеет особо важную роль, ведь при правильном построении стратегии развития бренда можно получить рост компании по всем показателям в многократном размере. Интернет способен обеспечить резкие скачки узнаваемости бренда в короткие промежутки времени, чего нельзя часто добиться в оффлайн-компаниях.

Процесс построения бренда в сети можно условно разбить на этапы:

1. анализ аудитории бренда, понимание ее портрета;
2. упаковка бренда;
3. формирование осведомленности о бренде;
4. укрепление позиций бренда;
5. усиление лояльности;
6. более эффективное взаимодействие с пользователями [3].

Благодаря широкому спектру инструментов, в интернете можно составить достаточной точный портрет потенциальной аудитории. Анализ ключевых метрик, определяющих местонахождение, возраст, интересы, сведения о деятельности позволит сформировать четкое понимание для упаковки бренда.

Упаковка бренда имеет особо важную роль в развитии интернет-стартапов. Под этим понятием понимаются аспекты, которые скорее будут выражать и представлять бренд в сети Интернет на различных площадках и в различных форматах (айдентика, нейминг, особенности коммуникации). Среди аспектов, характеризующих упаковку бренда интернет-стартапа, стоит выделить:

- логотип – должен быть уникальным, иметь несколько адаптивных версий для применения логотипа в качестве иконок, формирования доменного имени;
- название бренда – лучшая практика названия интернет-стартапа заключается в симбиозе распространенного понятия или фразы, раскрывающий область деятельности стартапа и его преимущества, а также добавление к названию этимологических, орфографических и визуальных особенностей. При таком симбиозе получится название стартапа, которое будет одновременно информативным и у всех «на слуху». Также очень популярны короткие запоминающиеся названия, основанные на несуществующих словах;
- особенности коммуникации – этот аспект важен при упаковке бренда интернет-стартапа, поскольку основой таких компаний является коммуникация на разных каналах. Если у интернет-стартапа есть так называемая политика общения с пользователями, то это будет сильно способствовать важному показателю «цитируемости в интернете» за счет которого достигается эффект сарафанного радио. Отличным примером выступает сервис по изучению английского языка LinguaLeo, который ведет всю коммуникацию с пользователями от лица вымышленного героя.

Все последующие этапы брендинга интернет-стартапов по большей части представляют из себя масштабирование каналов взаимодействия с пользователями.

Как правило, на этих этапах используются стандартные механизмы, которые применяются в брендинге как оффлайн-компаний, так и интернет-стартапов. Для формирования и продвижения бренда используется следующий инструментарий.

1. Имиджевая реклама в сети и оффлайне.
2. Спонсорство.
3. Партнерство.
4. Public Relations [3].

Пожалуй, главной особенностью брендинга интернет-стартапов является то, что бренд должен быть в постоянном развитии, ввиду коммуникативных возможностей среды. Бренд интернет-стартапов должен иметь легко меняющийся имидж, чтобы «подстройка» под конкретного пользователя заключалась лишь в смене сути рекламного сообщения, а не в

изменении сущности самого бренда. При этом суть бренда должна быть максимально доступной для понимания и иметь четкое позиционирование относительно других брендов.

В интернете, в сравнении со средой оффлайн-бизнеса, гораздо выше вероятность появления новых рынков. Кроме того, происходит непрерывный процесс изменения старых рынков, возникают новые ниши и возможности для бизнеса. Этот процесс идет очень быстро, не замедляясь ни на секунду. В каждом из сегментов идет ожесточенная борьба за пользователей, клиентов, имидж – т.е. за место на рынке.

В результате анализа методов брендинга интернет-стартапов, мы приходим к пониманию того, что для формирования и продвижения бренда интернет-проектов используются стандартные механизмы. Но на этапах его построения можно выделить ряд особенностей. Главной особенностью можно считать выбор каналов коммуникации с потенциальной аудиторией. А также создание адаптивного имиджа бренда, в связи с быстроменяющейся коммуникативной средой.

Литература

1. Годин А.М. Брендинг: учебное пособие. – М.: Дашков и К, 2013. – 183 с.
2. Успенский И. Энциклопедия Интернет-бизнеса. – СПб.: Питер, 2001. – 212 с.
3. Management.com [Электронный ресурс]. – Режим доступа: <http://www.management.com.ua/marketing/mark010.html>, своб.



Горбунов Антон Вячеславович

Год рождения: 1992

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: korvin_mckernek@mail.ru

УДК 004.92

АНАЛИЗ МЕТОДОВ РЕНДЕРИНГА МОДЕЛЕЙ В СИСТЕМЕ РЕНДЕРИНГА V-RAY ДЛЯ ДОСТИЖЕНИЯ ФОТОРЕАЛИСТИЧНОСТИ МОДЕЛЕЙ

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 «рендеринг», рассмотрены основные методы визуализации, используемые в трехмерной компьютерной графике, определены алгоритмы глобального освещения системы визуализации V-Ray и рассмотрены их основные достоинства и недостатки.

Ключевые слова: компьютерная графика, рендеринг, визуализация, глобальное освещение, V-Ray.

Термином «рендеринг» в компьютерной графике обозначается процесс визуализации трехмерных моделей или сцен, результатом которого является двухмерное растровое изображение. Выделяют две группы методов визуализации: методы рендеринга в режиме реального времени и методы предварительного рендеринга (или пре-рендеринга) [1, 2]. Обе группы различаются качеством получаемых изображений, степенью сложности и, соответственно, объемом временных затрат на выполнение визуализации моделей.

Рендеринг в реальном времени применяется в области компьютерных игр, в различных симуляторах, иными словами, везде, где имеет значение быстрая скорость обработки данных о трехмерных объектах и вывода получаемых изображений на экран.

Методы пре-рендеринга используются в кинопроизводстве и при создании анимационных фильмов, а также в проектировании архитектурных сооружений, дизайне интерьеров, в рекламе. В этом случае важную роль играет качество получаемых изображений и ценится высокая степень их реалистичности. При этом возможно добиться такой степени фотореализма изображений, что невозможно будет определить, является ли представленное изображение фотографией или это результат рендеринга (рис. 1).



Рис. 1. Примеры фотореалистичного рендеринга

Эффект фотореализма полезен при демонстрации проектов тем, что помогает в деталях воспроизвести вид возможного конечного результата и сделать презентацию более информативной.

Известны следующие методы визуализации [2, 3]:

1. метод растеризации – процесс геометрического проецирования моделей на плоскость, в результате которого получается растровое изображение. Растеризация содержит три этапа: тесселяцию (разбиение модели на примитивные полигоны); геометрическую обработку (трансформация, отсечение, освещение и т.д.); растеризацию (конвертирование двухмерного представления сцены или модели в растровое изображение);
2. метод «бросания» лучей (raycasting) – более медленный и, вместе с тем, более гибкий метод рендеринга, который заключается в том, что из одной точки наблюдения (камеры) испускаются лучи и в точках пересечения лучей с объектами сцены производится вычисление цвета пиксела на двумерном экране;
3. метод трассировки лучей – физически корректный метод рендеринга, в котором луч при пересечении с точкой на поверхности примитива не останавливает свое движение, а разделяется на три компонента: отраженный, преломленный и теневой лучи. Далее процесс рекурсивно повторяется. Количество таких повторов определяют глубину трассировки. Физический принцип, заложенный в данном методе, позволяет получить фотореалистичные изображения высокого качества;
4. метод трассировки пути, наиболее близкий к физическим законам распространения света, основан на применении метода Монте-Карло при решении уравнения рендеринга для вычисления освещенности, благодаря чему возможно воспроизведение множества оптических эффектов (построение теней, каустика и др.), получить которые с помощью других методик рендеринга либо затруднительно, либо вообще невозможно [4]. Трассировка пути позволяет получать фотореалистичные изображения, однако, этот метод является наиболее медленным по производительности.

На текущий момент известно достаточно большое число программ-рендеров. Наиболее популярными из них являются Arnold Renderer, Maxwell Renderer, RenderMan, Mental Ray, V-Ray, Corona Renderer и другие. Но среди них особенно выделяется V-Ray, как самый распространенный в русскоязычном пространстве высококачественный фотореалистичный визуализатор [1, 5]. Достоинством использования V-Ray является его недорогая стоимость и возможность

совместного использования с редактором 3dsMax, для которого существует бесплатная академическая лицензия.

В основе рендеринга V-Ray лежит метод трассировки лучей. Так, для того чтобы создать фотореалистичное изображение, процессору компьютера требуется просчитать пути как первичных лучей, так и отраженных, и теневых с заданной глубиной трассировки. При этом на процесс рендеринга влияют такие показатели, как количество источников света (с учетом вида этих источников), показатели преломления материалов объектов, присутствующих на сцене, типы отражений световых лучей от различных поверхностей, рассеивание лучей внутри объектов. Таким образом, V-Ray при рендеринге трассировкой лучей учитывает влияние объектов друг на друга, достигая при этом реалистичной имитации света. Алгоритмы, которые учитывают не только прямое освещение от источников света, но и отраженные лучи, называются алгоритмами глобального освещения (global illumination, GI) [1].

В V-Ray включены следующие алгоритмы глобального освещения [1, 4]:

1. прямое вычисление (Brute Force) – метод, при котором трассировка лучей происходит в два этапа: сперва определяются точки сцены, в которые попали выпущенные из камеры лучи, затем происходит расчет освещения с учетом трассировки первичных (или вторичных) отражений. С помощью данного алгоритма можно добиться хорошей детализации, но рендеринг сложных сцен может занять длительное время;
2. вычисление на основе карт свечения (светимости/освещения) – метод, позволяющий быстрее рассчитать глобальное освещение, но результат может быть уже не столь точным, как при прямом вычислении. Достоинством данного подхода является возможность сохранения карты свечения с целью ее повторного использования;
3. метод фотонных карт (Photon Map) – алгоритм, используемый для расчета вторичных отражений, заменяющий в ряде случаев более медленный подход прямого вычисления, однако этот способ расчета освещения требует большого объема памяти и не всегда справляется с обработкой мелких деталей. Фотонная карта представляет собой таблицу в кэш-памяти компьютера, хранящую данные, которые собираются на первом этапе работы алгоритма, – это информация о направлении фотонов, рассчитанная энергия частиц и точки пересечения фотонов с поверхностью объектов. На втором этапе рендеринга расчет освещения сцены производится с учетом данных фотонных карт. Этот метод используется в основном для приближенного расчета освещения сцены. Фотонные карты также можно использовать повторно;
4. последний алгоритм GI – световое кэширование (Light Cache). Данный алгоритм схож по своему принципу работы с методом фотонных карт, так как в этом случае тоже формируется карта глобального освещения, однако, трассируются лучи только из камеры, а не из всех источников света, как это устроено в алгоритме Photon Map. Одним из недостатков светового кэширования является то, что расчет освещения производится для конкретного положения камеры, но этот метод лучше справляется с визуализацией мелких деталей. Работа алгоритмов GI показана на рис. 2.

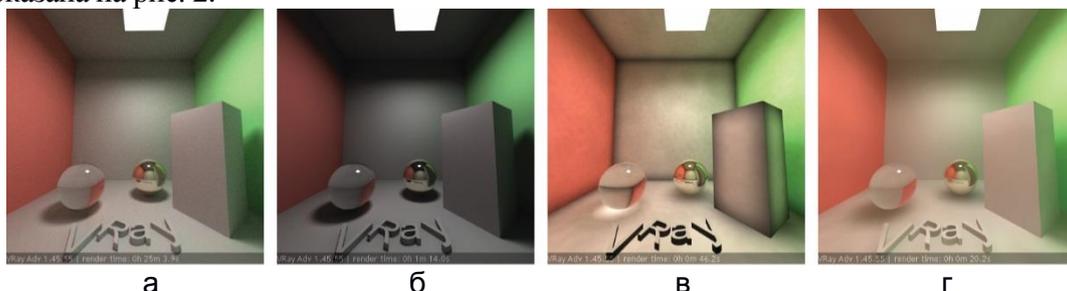


Рис. 2. Примеры работы алгоритмов расчета глобального освещения: brute force (4 отражения) (а); irradiance map (1 отражение) (б); photon map (без отражений) (в); light cache (без отражений) (г)

В работе были рассмотрены основные методы рендеринга, используемые в трехмерной компьютерной графике. Одним из самых популярных визуализаторов, используемых для получения фотореалистичных изображений, является система V-Ray, в основе которой лежит способ визуализации моделей с помощью трассировки лучей. Реалистичный результат может быть получен благодаря встроенным алгоритмам глобального освещения, каждый из которых имеет свои достоинства и недостатки.

В ходе дальнейшего исследования планируется определить критерии оптимальной оценки получаемых результатов рендера на основе проведения ряда тестов и далее, в соответствии с выявленными критериями, выдвинуть рекомендации по настройке системы V-Ray с целью получения фотореалистичных изображений приемлемого качества за оптимальное время рендеринга.

Литература

1. Романовский А.М. Визуализация сцены при помощи систем рендеринга V-Ray и Corona Renderer // Творчество молодых: дизайн, реклама, информационные технологии: сб. тр. XIV Межд. науч.-практ. конф. студ. и асп. – 2015. – С. 117–120.
2. Сесин И.Ю., Холкин И.И. Рендеринг – состояние и перспективы // Научный альманах. – – 2015. – № 10-3(12). – С. 233–236.
3. Вольхин К.А. Основы компьютерной графики: электронное учебное пособие для студентов направлений 270100 «Строительство» и 270300 «Архитектура» [Электронный ресурс]. – Режим доступа: http://ng.sibstrin.ru/wolchin/umm/1_kg/kg/index.htm, своб.
4. Методы GI [Электронный ресурс]. – Режим доступа: <http://vraydoc.narod.ru/vray220/gimethods.htm>, своб.
5. Попивчак И.И., Ходасевич Г.Б. Современное программное обеспечение для визуализации данных // Природные и техногенные риски (физико-математические и прикладные аспекты). – 2014. – № 1(9). – С. 58–64.



Горбунова Анастасия Александровна

Год рождения: 1988

Факультет методов и техники управления «Академия ЛИМТУ», кафедра управления и права, группа № S4117с

Направление подготовки: 38.04.02 – Менеджмент

e-mail: cherycavali@mail.ru

УДК 331.103.6

ВЛИЯНИЕ КРІ НА МОТИВАЦИЮ РАБОТНИКОВ

Горбунова А.А.

Научный руководитель – к.юрид.н., доцент Воронина М.Ф.

В работе описана разработка системы мотивации персонала на основе ключевых показателей (КРІ). Определены основные характеристики КРІ, описаны характерные черты стратегий ведения бизнеса, обозначено влияние стратегии на КРІ, рассмотрена модель расчета материальной мотивации на основе КРІ. На основании проведенного исследования сделаны выводы о влиянии КРІ на материальную и нематериальную мотивацию.

Ключевые слова: КРІ, ключевые показатели эффективности, стратегия, материальная мотивация, нематериальная мотивация.

Стратегии развития современных организаций предполагают постоянный рост объемов, повышение качества продукции и услуг, а, соответственно, повышение эффективности труда работников. При этом часто система оплаты труда предполагает выплату фиксированной суммы, которая не основана на повышении качества работы или количестве выполняемых операций, и поэтому тормозит развитие организации в целом. Для достижения стратегических целей организации возникает необходимость разработки инструментов и внедрение их в деятельность, а также разработки материальной мотивации персонала в зависимости от их личного вклада. KPI (Key Performance Indication) – ключевые показатели эффективности, которые позволяют оценить эффективность выполняемых действий предприятием в целом, отдельными подразделениями и каждым конкретным сотрудником. Таким образом, система мотивации, построенная на KPI, ориентирует работников на достижение максимально высоких личных и коллективных результатов.

Опрос около 1000 работодателей на предмет использования в деятельности KPI показал, что только около половины опрошенных российских компаний (49%) ввели KPI для оценки труда своих сотрудников. Многие из них установили KPI для части работников (23%). Тем самым усложнив достижение стратегической цели организации по следующим причинам: разный темп работы подразделений, которым установлены KPI и остальным; рассогласованность действий этих подразделений; формальное выполнение работы и отсутствие мотивации на результат в подразделениях без KPI. Перечисленные причины могут провоцировать конфликты в компании и являться демотивирующим фактором. Для того чтобы система мотивации на основе KPI работала, при ее разработке необходимо учесть следующее: система мотивации должна быть понятна работникам, в том числе и тем, которым не установлены KPI, но от них зависит результат; KPI должны быть достижимы, но не занижены; система мотивация должна быть направлена на повышение эффективности компании за счет повышения эффективности работников; система мотивации должна побуждать работников работать на результат и дать возможность влиять на материальное вознаграждение; разработанные KPI должны реализовывать стратегию роста компании. Иначе говоря, KPI – это набор показателей, отражающих аспекты организационной деятельности, которые наиболее важны в настоящий момент, и в будущем приведут к реализации основной цели компании. Дэвид Парментер выделяет 7 основных характеристик KPI:

1. нефинансовый характер. Предпочтительнее, чтобы измерялся не объем, а эффективность (т.е. количество встреч с клиентами, приносящими большую прибыль);
2. измерения отслеживаются часто. Это позволяет отслеживать ситуацию в режиме реального времени, а также дает менеджеру возможность выявить недобросовестных или неквалифицированных сотрудников;
3. требуют внимания со стороны топ-менеджера или генерального директора. Ответственные работники ежедневно отчитываются перед руководством;
4. требуют внимания и корректирующих действий со стороны всего персонала;
5. понимание ответственности каждым сотрудником и команды в целом;
6. влияют на наиболее критические факторы успеха. Другими словами, если внимание руководства сконцентрировано на KPI, то компания добивается успеха по всем направлениям;
7. оказывают позитивное воздействие на другие показатели [1].

Разработка системы мотивации на основе KPI имеет следующую последовательность действий: определяется область измерения (финансы, производство, развитие, рынок и др.); осуществляется распределение общих целевых показателей на уровни управления компанией; распределяются роли на всех уровнях управления компанией, выделяются управляющие, которые несут ответственность за достижение результата своей команды; определяются ключевые показатели эффективности деятельности каждого управляющего и для членов его команды; определяется стоимостная оценка показателей в системе материальной мотивации; определяется зависимость системы материальной мотивации от достижения ключевых показателей эффективности.

KPI в первую очередь зависят от целей и стратегии компании. Майкл Портер выделяет три типа стратегии в бизнесе: «Лидерство по издержкам», «Стратегия дифференциации», «стратегия концентрации» (рыночной ниши). Для компаний, выбравших за основу стратегию «Лидерство по издержкам» характерно стремление сократить расходы на производство. Реализация данной стратегии предполагает крупные инвестиции в организацию масштабного производства, экономию за счет широкого производства и строгий контроль за расходами (постоянный контроль качества, стандартизация, эффективность производственных процессов). «Стратегия дифференциации» предполагает создание уникального продукта, и как следствие, развитие лояльности к торговой марке у потребителей: высокое качество, функциональность, высокие затраты на продвижение оправдывают высокую цену продукции [2]. При выборе «Стратегии концентрации» компания ориентируется на специфический сегмент рынка через низкие цены или уникальное предложение. Для данной стратегии характерно комплексное решение для потребителей, личные контракты и долгосрочные отношения, широкий спектр компетенций работников и программы лояльности [3].

Также выбранная стратегия оказывает влияние на набор KPI и частоту их измерения. Стратегия «Лидерство по издержкам» требует постоянного контроля за издержками на всех этапах производства, отслеживания скорости ключевых процессов, оборачиваемости активов и узнаваемости бренда. Показателями «Стратегии дифференциации» будет доля брака, объем продаж новых товаров, рентабельность продаж. Компании со «Стратегией концентрации» будут отслеживать индекс удовлетворенности клиентов, объем продаж постоянным клиентам, процент выполнения индивидуальных заказов, число претензий.

Ключевые показатели эффективности должны быть в балансе, но на практике финансовым показателям уделяется большее внимание, чем нефинансовым, так же, как и количественным – чем качественным, а индивидуальным – чем командным и т.д. Состав показателей зависит от целей, особенностей и стратегии компании, но баланс по различным видам показателей необходимо поддерживать.

После создания системы измерения показателей, разрабатывается система вознаграждения [4]. Стандартная формула расчета заработной платы имеет вид:

Заработная плата = оклад + премия.

При расчете премии с применением KPI формула заработной платы примет следующий вид:

Заработная плата = Оклад + Плановая сумма премии (Коэффициент $KPI_1 \cdot$ Коэффициент $KPI_2 \cdot$ Коэффициент KPI_n).

Исходя из целей компании, для каждого показателя может быть установлен вес. Вес оказывает влияние на премию и может быть различным. В случаях, когда для KPI установлен вес, тогда расчет принимает следующий вид:

Заработная плата = Оклад + Плановая сумма премии (Коэффициент $KPI_1 \cdot$ Вес $KPI_1 +$ Коэффициент $KPI_2 \cdot$ Вес $KPI_2 +$ Коэффициент $KPI_n \cdot$ Вес KPI_n).

Когда целесообразно применять усредненный KPI, то сумма коэффициентов KPI с учетом веса каждого из показателей, делится на количество учитываемых показателей. Коэффициенты устанавливаются в зависимости от достижения установленных KPI. Например: при достижении установленных значений на уровне 80% от планируемого результата, работнику начисляется минимальная премия. При понижении эффективности ниже уровня 80%, работнику выплатят только оклад. Максимальный уровень заработной платы будет выплачен работнику при достижении 120% установленных KPI. Приведенный пример является распространенной схемой расчета материальной мотивации на основе KPI, но компании самостоятельно устанавливают необходимые значения коэффициентов в зависимости от своих стратегических целей.

Описанная выше система материальной мотивации персонала на основе КРІ также направлена и на повышение нематериальной мотивации работников. Согласно основным процессуальным теориям мотивации, а именно теории Портера–Лоулера, теории ожиданий В. Врума и теории справедливости С. Адамса, положительное влияние на мотивацию оказывают следующие факторы: понимание результата; справедливость вознаграждения; осознание важности своей работы; наличие обратной связи от руководителя; признание в команде; автономия в решении определенного круга вопросов; разнообразие навыков; самореализация [5]. Система мотивации на основе КРІ основывается на управлении по целям, обеспечивая тем самым работникам понимание целей компании и роль каждого члена команды. Система материальной мотивации прозрачна и позволяет работнику самостоятельно определить, на каком этапе достижения цели он находится, а также размер своего вознаграждения. Система измерений и технология расчетов объективно показывает руководителю результаты работы каждого работника и подразделения в целом, что позволяет контролировать процессы и оперативно реагировать на них, давать обратную связь и нести ответственность за результат.

Таким образом, система КРІ для конкретной компании разрабатывается в соответствии с выбранной стратегией, особенностями производства и ведения бизнеса. Главная задача построения системы материальной мотивации на основе КРІ – выработка оптимальных значений показателей, важных, достижимых и понятных персоналу. Показатели требуют постоянного контроля со стороны руководства, а в случае изменений во внешней или внутренней среде, КРІ могут быть изменены или заменены. В противном случае КРІ будут демотивировать работников, и эффективность труда будет падать [6].

Литература

1. Парментер Д. Ключевые показатели эффективности. Разработка, внедрение и применение решающих показателей / Пер. с англ. А. Платонова. – М. ЗАО «Олимп-Бизнес», 2008. – 288 с.
2. Альстранд Б., Минцберг Г., Лампель Ж., Стратегическое сафари: Экскурсия по дебрям стратегического менеджмента / Пер. с англ. – М.: Альпина Паблишер, 2013. – 367 с.
3. Котлер Ф., Келлер К.Л., Маркетинг менеджмент. Экспресс курс/ Пер. с англ. под науч. ред. С.Г. Жильцова. –3-е изд. – СПб.: Питер, 2016. – 480 с.
4. Попов В.Л., Марков Д.А., Гуреева Е.Г., А.В. Крутова, Управление производством и операциями: учебное пособие. Стандарт третьего поколения. – СПб.: Питер, 2014. – 336 с.
5. Шаховой В.А., Шапиро С.А., Мотивация трудовой деятельности: учебное пособие. – 4-е изд. – М.-Берлин: Директ – Медиа, 2015. – 425 с.
6. Портер М. Конкурентная стратегия: Методика анализа отраслей и конкурентов / Пер. с англ. – 6-е изд. – М.: Альпина Паблишер, 2016. – 600 с.



Гуляева Анастасия Валерьевна

Год рождения: 1992

Факультет методов и техники управления «Академия ЛИМТУ»,
компьютерного проектирования и дизайна, группа № S4206

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: mail.gulyaeva@ya.ru



Готская Ирина Борисовна

Год рождения: 1957

Факультет методов и техники управления «Академия ЛИМТУ»,
компьютерного проектирования и дизайна, д.педагог.н., профессор
e-mail: iringot@mail.ru

УДК 004.51

ИССЛЕДОВАНИЕ ИНСТРУМЕНТОВ АНАЛИЗА ЮЗАБИЛИТИ САЙТОВ

Гуляева А.В.

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

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе рассмотрены понятие юзабилити, web-юзабилити и этапы юзабилити-тестирования с описанием каждого шага тестирования.

Ключевые слова: юзабилити, юзабилити-тестирование.

Юзабилити – степень, с которой продукт может быть использован определенными пользователями при определенном контексте использования для достижения определенных целей с должной эффективностью, продуктивностью и удовлетворенностью [1].

Web-юзабилити нацелено на то, чтобы сайт был коммерчески успешен, т.е. приносил заявки и продажи. Таким образом, понятие web-юзабилити сайта можно еще трактовать как удобство совершения конверсионного действия при попадании пользователя на ресурс [2].

Сайт, учитывающий правила web-юзабилити, лучше воспринимается пользователем, имеет хороший показатель времени, проведенного на сайте, и высокий процент конверсии. Все вместе приводит к тому, что пользовательские факторы начинают работать на сайт, принося клиентов и повышая позиции сайта в поисковых системах.

По мнению Стива Круга первое и главное правило юзабилити – «не заставлять посетителя сайта думать», и это правило является решающим при оценке того, что и как работает (или не работает) в веб-дизайне. Этот принцип означает, что веб-страница должна быть максимально простой и понятной [3].

Работу по оценке юзабилити сайта следует начинать уже при создании сайта, продумывая каждую мелочь. Если же сайт уже давно работает, но хочется получить от него большую отдачу, то также следует провести анализ юзабилити.

Работу по аудиту и анализу юзабилити можно разделить на несколько этапов.

1. Сбор данных. Первым шагом к пониманию того насколько продукт соответствует ожиданиям пользователей – это сбор статистических данных. Для этого есть специальные счетчики. Самые популярные из них – это Google Analytics, Яндекс.Метрика, Openstat, Liveinternet. С их помощью можно отслеживать показания счетчиков отказов, количество просмотров страниц, источники трафика, ключевые слова, географию посетителей, целевые страницы, страницы выхода. Данные предоставляются бесплатно – таким образом, это один из самых доступных вариантов анализа юзабилити.

Если необходимо получить более конкретную информацию, то можно также подключить систему отслеживания кликов пользователей («тепловая карта»). Это позволит узнать, что именно привлекает пользователей, и отследить их путь по сайту.

2. Оценка данных. Есть разные подходы к оценке результатов:

- фокус-группы;
- эвристической оценки;
- индивидуальное интервью;
- экспертные оценки и т. д.

Каждый из этих подходов имеет свои плюсы и минусы, их надо применять в зависимости от поставленной задачи.

Проанализировав результаты юзабилити-тестирования можно определить требования к интерфейсу для перепроектирования.

3. Внедрение и проверка выводов. После сбора статистики и оценки данных необходимо начать внедрять изменения на сайте. Изменения могут быть в логике работы функциональных элементов, дизайне, текстах и их стилистике, графических элементах и их анимации.

Также для оценки юзабилити сайта можно заказать экспертизу у профессионалов. Эта услуга является платной, но результат может превзойти ожидания. Другой вариант – провести экспертизу самостоятельно. Для этого необходимо иметь опыт и навыки поиска и анализа юзабилити-ошибок. В этом могут помочь такие онлайн-сервисы, как Usability Hub, Usabilla, Optimal Workshop и др. Сервисы анализируют структуру сайта, дизайн, а также предоставляют тестирование с реальными пользователями. Некоторые из этих сервисов также являются платными, но это будет дешевле, чем заказывать экспертизу в специализированной компании.

Юзабилити во многом базируется на человеческой психологии и человеческих привычках. Зная стандарты в области построения сайтов, а также инструменты и методы для анализа юзабилити можно быстро обнаружить ошибки в удобстве использования ресурса.

Литература

1. ГОСТ Р ИСО 9241-210-2012. Эргономика взаимодействия человек-система. Часть 210. Человеко-ориентированное проектирование интерактивных систем. – Введен 01.12.2013. – М.: Стандартинформ, 2013. – 32 с.
2. Аудит юзабилити сайта [Электронный ресурс]. – Режим доступа: <http://www.sembook.ru/book/povyshenie-konversii-sayta/yuzabiliti-sayta/>, своб.
3. Krug S. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability. – Published by Que, 2014. – 216 p.

Гумбург Виктория Геннадьевна

Год рождения: 1993

Факультет информационной безопасности и компьютерных технологий,
кафедра проектирования и безопасности компьютерных систем, группа № P4255

Направление подготовки: 10.04.01 – Информационная безопасность

УДК 004

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АНТИВИРУСНЫХ ПРОГРАММ ДЛЯ МАЛОГО И СРЕДНЕГО БИЗНЕСА

Гумбург Б.Ю., Гумбург В.Г.

Научный руководитель – к.т.н., профессор Мартынов В.П.

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

В работе рассмотрены задачи антивирусной защиты локальных сетей. Принципы построения защиты. Также произведено сравнение антивирусных средств.

Ключевые слова: антивирус, локальная сеть, средства защиты, Dr.Web, Kaspersky, Eset Nod32.

Введение. Вопросы эффективной антивирусной защиты сегодня как никогда актуальны и в корпоративном секторе, и среди частных пользователей, однако, в отличие от последних, проблемы и задачи, встающие перед компаниями, намного серьезнее и требуют решений иного уровня. Администраторам корпоративных информационных систем приходится устанавливать антивирусные средства, конфигурировать их и настраивать политики обновления, а также следить, чтобы антивирусы постоянно были включены на сотнях или даже тысячах машин, – и зачастую делать все это приходится вручную [1, 2].

Антивирусная защита и ее принципы. Задача антивирусной защиты актуальна сегодня как для частных пользователей, так и для компаний. Для компаний эти вопросы наиболее трудоемки, так как требуют решения на ином уровне, нежели у частных пользователей. Администраторам необходимо устанавливать, конфигурировать, обновлять, следить за правильной работоспособностью антивирусных средств на большом количестве рабочих компьютеров предприятия.

Заражение даже одного компьютера в сети несет опасность всей локальной сети. Заражение может распространиться на другие компьютеры сети, например, через заражение файлов сервера, общих документов, передачу по корпоративной почте. В этой связи все устройства, подключаемые к сети, должны быть оснащены антивирусным программным обеспечением. Но также необходимы и организационные меры.

Система антивирусной защиты может строиться по нескольким принципам:

- принцип реализации единой технической политики при обосновании выбора антивирусных продуктов для различных сегментов локальной сети;
- принцип полноты охвата системой антивирусной защиты всей локальной сети организации;
- принцип непрерывности контроля локальной сети предприятия, для своевременного обнаружения компьютерной инфекции;
- принцип централизованного управления антивирусной защитой.

1. Принцип реализации единой технической политики. Данный принцип предполагает использование только того программного обеспечения, которое рекомендовано специалистом по информационной безопасности. Такая политика ориентирована на долгосрочное использование, позволяет хорошо планировать затраты на антивирусное программное обеспечение и его последующее обновление.
2. Принцип полноты охвата системой антивирусной защиты всей локальной сети организации. Предусматривает постепенное внедрение программных средств антивирусной защиты до полного оснащения в сочетании с организационно-режимными мерами защиты информации.
3. Принцип непрерывности контроля локальной сети предприятия. Организация защиты сети должна обеспечивать возможность проверки состояния сети в любое время.
4. Принцип централизованного управления антивирусной защитой. Контроль за антивирусными средствами должен осуществляться из одного органа. Куда также должны поступать журналы работы сети, доклады пользователей об обнаружении вирусов, а также управление системой антивирусной защиты.

Таблица. Сравнение средств защиты

	Eset Nod32 Endpoint Security 6	Dr. Web «Малый бизнес»	Kaspersky Endpoint Security для бизнеса «Стандартный»
Стоимость	11813 рублей	4 990 рублей	17460 рублей (минимально 10 лицензий)
Срок лицензии	1 год	1 год	1 год
Централизованное управление	+	+	+
Сетевой экран	+	+	+
Степень защиты (баллов)	3	5	5
Нагрузка на ПК (баллов)	5	4	3
Обновления баз (баллов)	5	5	5
Техническая поддержка (баллов)	5	5	5
Сумма баллов	18	19	18

Числовые оценки выставлены по 5-ти бальной системе.

Заключение. Из таблицы мы видим, что по суммарным результатам лидирует Dr. Web, при этом имеет наименьшую стоимость. Заметим, что Kaspersky Endpoint Security для бизнеса «Стандартный» продается минимум на 10 рабочих станций, чем и обусловлена высокая цена.

Литература

1. Филин С.А. Информационная безопасность. Учебное пособие. – М.: Альфа-Пресс, 2006. – 412 с.
2. Сычев Ю.Н. основы информационной безопасности: учебно-практическое пособие. – М.: Изд. центр ЕАОИ, 2007. – 300 с.



Давлетшин Владислав Равилевич

Год рождения: 1995

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4108

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: vlad35066@mail.ru

УДК 004.4

ОБЗОР КОМБИНАЦИЙ ФРЕЙМВОРКОВ И ДРУГИХ ТЕХНОЛОГИЙ ДЛЯ СОЗДАНИЯ ВЕБ-ПРИЛОЖЕНИЙ

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 приведен обзор некоторых эффективных комбинаций фреймворков для разработки веб-приложений и исследованы пути их взаимодействия.

Ключевые слова: веб-приложение, фреймворк, стек технологий.

Фреймворки – это программные продукты, которые упрощают создание и поддержку технически сложных или нагруженных проектов. Фреймворк, как правило, содержит только базовые программные модули, а все специфичные для проекта компоненты реализуются разработчиком на их основе. Тем самым достигается не только высокая скорость разработки, но и большая производительность и надежность решений. Фреймворк можно назвать платформой, которая подходит для создания сайтов, бизнес-приложений и веб-сервисов [1–3].

Зачем же они собственно нужны? Допустим появилась задача, в ходе которой необходимо реализовать веб-сервис или веб-приложение. Если выбрать способ писать все самостоятельно с нуля, не используя фреймворки, тогда в ходе разработки вы столкнетесь с огромным количеством проблем таких как неоптимальное/неграмотное написание кода и прочими, а можно тщательно подобрать фреймворки, с помощью которых реализация будет быстрее и надежнее. Сейчас веб-приложения часто используют стандартизированную структуру организации компонентов, а создание структуры при использовании фреймворков значительно упрощается. Также перед веб-разработчиками часто стоит выбор между фреймворками и коробочными CMS.

Рассмотрим проект, реализуемый с помощью фреймворков:

Плюсы:

- в большинстве случаев не уступают CMS-системам в производительности, адаптивности и выдерживают большую нагрузку;
- легок в сопровождении;
- можно реализовать любой бизнес-процесс;
- проект можно легко масштабировать и модернизировать.

Минусы:

- объем работы, как правило, больше, чем в случае с использованием коробочной CMS;
- необходимо детальное понимание протекающих бизнес-процессов.

Теперь рассмотрим несколько популярных front-end фреймворков:

- AngularJS – JavaScript-фреймворк с открытым исходным кодом. Предназначен для разработки одностраничных приложений. Фреймворк работает с HTML, содержащим дополнительные пользовательские атрибуты, которые описываются директивами, и связывает ввод или вывод области страницы с моделью, представляющей собой обычные переменные JavaScript;
- Bootstrap – свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения. Бутстрап сейчас на пике популярности, и можно уверенно сказать, что на его основе можно сделать любой веб-интерфейс;
- Ember.js – свободный JavaScript каркас веб-приложений, реализующий MVC-шаблон, предназначенный для упрощения создания масштабируемых одностраничных веб-приложений.

Рассмотрим несколько популярных back-end фреймворков:

- Node.js – серверная реализация языка программирования JavaScript, основанная на движке V8. Предназначена для создания масштабируемых распределенных сетевых приложений, таких как веб-сервер;
- Yii – это высокоэффективный, основанный на компонентной структуре PHP-фреймворк для быстрой разработки крупных веб-приложений. Он позволяет максимально применить концепцию повторного использования кода и может существенно ускорить процесс веб-разработки.

Теперь рассмотрим несколько связок фреймворков:

- AngularJS и NodeJS. Эти два фреймворка активно используются в разных стеках технологий, самый известный из них пожалуй «MEAN». MEAN – это аббревиатура, под которой

объединены три фреймворка JavaScript и документно-ориентированная NoSQL-база данных: M – это MongoDB, E – это ExpressJS, A – это AngularJS, N – это NodeJS. К плюсам можно отнести то, что оба фреймворка работают на JavaScript, это дает возможность разработчику со знанием языка полностью заниматься и клиентской и серверной стороной. В конечном итоге быстрее и проще масштабировать. MongoDB представляет собой базу данных NoSQL, которая оптимизирует запросы. Использование JavaScript также улучшает развитие, потому что один язык применяется и для внешней и внутренней работы;

- Bootstrap и Yii. Для Yii написано множество расширений, позволяющих использовать Yii вместе с CSS-фреймворком Bootstrap. Самые известные – это Yiistrap и набор дополнительных виджетов Yii Wheels. Если взять шаблон, написанный с использованием Bootstrap, то программисты без помощи дизайнеров и верстальщиков смогут сразу создать отличный интерфейс для управления системой. Все элементы уже созданы, нужно просто брать их и собирать, как дом из кирпичиков. В итоге мы получили мощный, производительный инструмент, с которым могут работать все программисты. Кстати, по тестам на производительность Yii показывает весьма высокие результаты;
- EmberJS и NodeJS. Не слишком популярная связка фреймворков, но от того не менее производительная, как в случае с Angular – это отличное решение для разработки веб-приложений. Благодаря хорошему обработчику путей и опциональному слою для работы с данными под названием «ember data» работа в связке с NodeJS значительно упрощается, но встроенный шаблонизатор Handlebars, который использует Ember, загрязняет DOM множеством тегов, что не только усложняет HTML, но и может поломать CSS или интеграцию с другими фреймворками, такими как jQuery UI Sortable.

Заключение. В конечном итоге выбор стека технологий и решение использовать коробочные CMS или фреймворки стоит за командой разработчиков. В приведенных выше примерах комбинаций фреймворков показано, каких результатов можно добиться разными технологиями. Выбор стека технологий – это отдельная обширная тема, включающая в себя анализ предполагаемой аудитории, выбор основной платформы и многое другое. В данной работе упущены некоторые широко использованные языки, такие как Ruby или Python и сделан акцент на JavaScript. На данный момент AngularJS+NodeJS – самая популярная связка фреймворков, которые используются для создания веб-приложений, которая гарантирует вам наибольшую безопасность и удобство в разработке.

Литература

1. AngularJS:Miscellaneous:Develop [Электронный ресурс]. – Режим доступа: <https://docs.angularjs.org/misc/contribute>, своб.
2. Ember.js-Ember2.11 [Электронный ресурс]. – Режим доступа: <http://emberjs.com/blog/2017/01/23/ember-2-11-released.html>, своб.
3. Рейтинги front-end и back-end фреймворков [Электронный ресурс]. – Режим доступа: <http://tagline.ru/backend-frontend-frameworks-rating/>, своб.



Давыдова Анастасия Сергеевна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107c

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: davydova_anastasia@list.ru



Сокуренко Юрий Андреевич

Год рождения: 1937

Факультет методов и техники управления «Академия ЛИМТУ»,

к.т.н., доцент

e-mail: kpd@limtu.ru

УДК 004

ОБЗОР СОСТОЯНИЯ РЫНКА ОФИСНЫХ ОНЛАЙН-ПАКЕТОВ

Давыдова А.С., Сокуренко Ю.А.

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 приведены наиболее популярные на текущий момент офисные онлайн-пакеты. Представлено краткое описание каждого из них.

Ключевые слова: офисный онлайн-пакет, онлайн-офис, SaaS, Google Docs, Office Online, Feng Office, МойОфис, ThinkFree Online, ONLYOFFICE, Zoho Docs.

Развитие электронного документооборота и программного обеспечения как услуги (SaaS) привели к появлению понятия – офисный онлайн-пакет. Офисные онлайн-пакеты реализуются на базе модели обслуживания SaaS.

SaaS – модель (Software-as-a-Service, программное обеспечение как услуга), в которой потребителю предоставляется возможность использования прикладного программного обеспечения (ПО) провайдера, работающего в облачной инфраструктуре и доступного из различных клиентских устройств или посредством тонкого клиента, из браузера (например, веб-почта) или интерфейс-программы [1].

Офисные онлайн-пакеты представляют собой набор веб-сервисов, например, таких как текстовый редактор, электронные таблицы, создание презентаций и т.д. Офисный онлайн-пакет включает все основные возможности традиционных офисных пакетов, но в отличие от традиционных офисов доступны с любого компьютера, имеющего доступ в Интернет. С появлением онлайн-офисов, у пользователей появилась возможность совместно работать над проектами в любое время, находясь при этом в любой точке земного шара. Что касается стоимости, то обычно базовый набор веб-сервисов бесплатен, а за расширенную версию необходимо платить абонентскую плату.

На сегодняшний день IT-рынок ПО предлагает следующие офисные онлайн-пакеты:

1. Google Docs;
2. Office Online;
3. Feng Office;
4. МойОфис;
5. ThinkFree Online;
6. ONLYOFFICE;
7. Zoho Docs.

Рассмотрим подробнее каждый из них.

Google Docs – бесплатный офисный онлайн-пакет, в состав которого входят: текстовый редактор, табличный процессор, сервис для создания презентаций, а также сервис для работы с формами [2]. Образован в результате слияния компонента Writely (компании Upstrtle) и Google Spreadsheets (компании 2Web Technologies). Помимо поддержки доступа

со стационарного компьютера, компания Google занимается разработкой приложений для мобильных платформ Android и iOS.

Кратко рассмотрим компоненты Google Docs:

- Writely – текстовый процессор, основанный на AJAX, позволяющий создавать и редактировать текстовые документы. Есть возможность работы в офлайн, при этом документ запишется при первом подключении к интернету. Обладает достаточным количеством средств форматирования: смена размера и стиля шрифта, выбор цвета и декораций, создание списков и таблиц, вставка картинок, ссылок и специальных символов. Поддерживаются следующие форматы: простой текст, pdf, rtf, OpenDocument, Microsoft Word, html и несколько графических форматов;
- Google Spreadsheets – табличный процессор, позволяющий создавать электронные таблицы, а также производить несложные вычисления. Позволяет строить цветные графики и диаграммы;
- Google Presentations – приложение для создания и работы с презентациями. Имеет множество встроенных тем, шрифтов, видео, видов анимации;
- Google Forms – приложение, позволяющее провести опрос, составить список, собрать адреса электронной почты для новостной рассылки и провести викторину [3].

Office Online – офисный онлайн-пакет, позволяющий пользователям создавать и редактировать файлы, используя облегченные веб-версии приложений Microsoft Office: Word, Excel, PowerPoint, OneNote. Помимо этих приложений в пакет входят Sway, Outlook.com, OneDrive, приложения Люди и Календарь. Корпоративная версия сервиса под названием Office Web Apps Server может быть установлена на приватные облачные хранилища вместе с такими сервисами, как Microsoft SharePoint, Microsoft Exchange Server и Microsoft Lync Server [3].

Для некоммерческого использования Office Online бесплатен, для этого необходимо зайти с помощью аккаунта Microsoft на сайт Office.com, который также совмещен с сервисом OneDrive и Office Live Workspace. Корпоративные версии доступны через сервис Office 365.

Feng Office – офисный онлайн-пакет с открытым исходным кодом, разработанный Feng Office. Распространяется по лицензии AGPL. Существует как бесплатная версия, которая включает базовый набор веб-сервисов, так и платные с расширенными возможностями. Максимальная версия включает в себя такие компоненты как управление документами, проектами, клиентами и контактами, создание документов, заметок, календарей, табеля, управление контактами, поддерживает форумы и Wiki и др. Используются следующие технологии: PHP, JavaScript, AJAX, MySQL. Рекомендуемые требования к серверу: Apache HTTP Server 2.0, PHP 5.4, MySQL 5 (с поддержкой InnoDB). При этом клиенту достаточно иметь современный веб-браузер [4].

МойОфис – пакет офисных приложений, разрабатываемый российской компанией «Новые облачные технологии».

Состоит из следующих компонентов:

1. документы: текст – редактор для работы с текстовыми документами; таблица – редактор для построения электронных таблиц и ведения экономико-статистических расчетов; презентация – редактор с полным набором инструментов для просмотра графических презентаций;
2. почта – приложение для организации и ведения внутренней и внешней переписки по электронной почте;
3. календарь – приложение для составления расписания мероприятий и планирования рабочих процессов;
4. контакты – приложение для хранения, управления и синхронизации адресной книги пользователя;
5. хранилище – продукт для создания централизованного хранилища данных в государственных организациях и крупных коммерческих предприятиях.

Преимущества МойОфис:

- поддержка всех популярных современных платформ: Windows (начиная с XP), macOS, Linux, а также iOS, Android и Tizen;
- возможность работы на операционных системах Windows (начиная с XP), macOS, ALT Linux и Astra Linux;
- единый интерфейс на всех платформах [5].

ThinkFree Online – онлайн-офис, включающий в себя текстовый и табличный редактор, а также приложение для создания презентаций. Реализован с применением технологий Java и AJAX. Поддерживает все виды форматов MS Office 2003 и 2007. Представляет собой схожий с MS Office интерфейс, что помогает легко ориентироваться пользователям. Поддерживает операционные системы такие как Windows, Mac и Linux, а также мобильные платформы Android, iOS и Windows Phone. Для работы через веб-браузер необходима предварительная установка Java Runtime Environment (JRE).

ONLYOFFICE – офисный онлайн-пакет, состоящий из такого набора приложений как онлайн-редактор документов, система управления документацией, инструмент для управления почтой, набор инструментов для менеджера по продажам, управление проектами (CRM), инструмент для управления временем (календарь) и инструменты для построения корпоративной сети. Онлайн-редакторы написаны на HTML5 Canvas. Поддерживают все популярные форматы, такие как: PDF, TXT, DOCS, DOC, ODT, RTF, HTML, EPUB, XLS, XLSX, ODS, CSV и PPTX. Онлайн-редакторы включают в себя полнофункциональный текстовый онлайн-редактор, который гарантирует 100%-ю идентичность при просмотре документов; редактор электронных таблиц для автоматизации вычислений и анализа большого объема данных; редактор презентаций, а также возможность расширения функциональных возможностей редакторов при помощи сторонних дополнений. Поддерживает такие платформы как Windows, Mac, Linux, Android и iOS.

Zoho Docs – офисный онлайн-пакет, состоящий из редактора для создания и редактирования текстовых документов (Zoho Writer), электронных таблиц (Zoho Sheet) and презентаций (Zoho Show). Поддерживает форматы: DOC(X), ODT, RTF, TXT, HTML, PDF, XLS(X), SXC, ODS, CSV, TSV, PPT, PPS, ODP, SXI, PPTX, PPSX. Поддерживает как десктопные платформы, так и мобильные. Существует как бесплатная версия, так платные standard (30ГБ), pro (100ГБ) и enterprise (1ТБ) за 2, 5 и 8 долларов за 1 пользователя за месяц соответственно.

Несомненно, офисные онлайн-пакеты не заменяют полностью традиционные офисные пакеты, но имеют существенный плюс – возможность работать над документами одновременно нескольким пользователям в режиме реального времени, имея при этом только доступ к сети Интернет, тем самым способствуя повышению мобильности доступа к различной документации пользователям. А так как онлайн-офисы основаны на модели обслуживания SaaS, то они позволяют снизить затраты на установку, обновление и поддержку ПО.

Литература

1. Баранова С.С. Исследование тенденций развития облачных сервисов // Cloud of science. – 2014. – Т. 1. – № 3. – С. 517–523.
2. Батура Т.В., Мурзин Ф.А., Семич Д.Ф. Облачные технологии: основные модели, приложения, концепции и тенденции развития // Программные продукты и системы. – 2014. – № 3(107). – С. 64–72.
3. Лучшие бесплатные онлайн-офисы для работы с документами. [Электронный ресурс]. – Режим доступа: <http://ocompah.ru/luchshie-besplatnye-onlajn-ofisy.html>, своб.
4. Официальный сайт Feng Office [Электронный ресурс]. – Режим доступа: <http://www.fengoffice.com/>, своб.
5. Официальный сайт МойОфис [Электронный ресурс]. – Режим доступа: <https://myoffice.ru/>, своб.



Дерманов Марк Константинович

Год рождения: 1992

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: mark@dermanov.ru

УДК 004.438

СРАВНИТЕЛЬНЫЙ ОБЗОР АВТОМАТИЗИРОВАННЫХ СИСТЕМ ДЛЯ УЧЕТА ЗАДАЧ ПРИ РАЗРАБОТКЕ ВЕБ-ПРОЕКТОВ

Дерманов М.К.

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

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В данном сравнительном обзоре рассмотрены наиболее популярные автоматизированные системы для учета задач при разработке интернет-проектов и наиболее важные функциональные возможности, которыми должны обладать подобные системы.

Ключевые слова: интернет-проекты, сайты, веб-сайты, автоматизированные системы, разработка сайтов, поддержка интернет-проектов, учет задач по проекту.

Современные интернет-проекты – сложные информационные системы. Состоящие из множества отдельных модулей, соединенных между собой. Между ними передается информация и они зависят друг от друга. Чтобы вся система работала без сбоев требуется корректная работа каждого отдельно взятого модуля [1–3].

Над разными участками проекта могут работать разные люди. При этом при объединении готовых модулей в одну систему необходимо, чтобы вся система работала, как было задумано. Для этого при перед этапом разработки, на этапе формирования требований к системе, формируются требования сначала ко всей системе в целом, а потом вся система разбивается на отдельные логически и(или) функциональные модули. Между модулями описываются способы взаимодействия, входные и выходные параметры.

Разработка каждого отдельного модуля состоит из множества связанных подзадач. У каждой задачи есть свое описание, список требований для реализации и прочая детальная информация по задаче, в том числе ответственные и сроки выполнения задачи. Даже в рамках одного модуля, разные задачи могут выполняться разными людьми. Над одной задачей может последовательно работать несколько человек, каждый на своем этапе.

Большое количество задач, сложная зависимость между ними, большое количество участвующих людей, и необходимость строгого соблюдения сроков и последовательности выполнения задач – послужили причиной для создания автоматизированных систем (АС) для учета задач при разработке интернет проектов.

АС подобного рода способна взять на себя всю тяжесть структурирования и упорядочивания информации. Использование АС позволяет осуществлять быстрый поиск нужной информации. Вся информация храниться в одном централизованном хранилище. Благодаря этому, все участники системы всегда могут видеть актуальную информацию на текущий момент. Ситуация, когда «данные устарели» при таком подходе исключена.

АС для учета задач – это многопользовательская система, в которой каждый пользователь видит только необходимый ему срез данных из всей АС. Участники системы могут иметь разные уровни прав. Руководители подразделений могут ставить и наблюдать за

ходом выполнения поставленных задач своим подчиненным. Подчиненные могут видеть только задачи, в которых они являются ответственными.

В ходе работы над задачей могут появиться различные вопросы для обсуждения, что и как следует делать. Это могут быть вопросы, связанные с неточностью описания задачи с технической точки зрения, это могут быть различные способы решения одной и той же задачи. Необходимость обсудить что-либо между участниками может породить большое количество причин.

Без использования подобной АС, у участников проекта, исключая телефон и почту, как правило, остается один рабочий канал для обсуждения всех вопросов – электронная почта. Общение по электронной почте имеет множество преимуществ, таких как быстрая скорость доставки сообщений, возможность ставить всех связанных с решением данного вопроса участников в список получателей письма.

Есть ряд существенных проблем, которые появляются при использовании электронной почты для обсуждения текущих вопросов (так как вся информация хранится где-то в электронной почте участников обсуждения). Выделим несколько из них:

- нет возможности привязать обсуждения к задаче;
- быстро ознакомится со списком вопросов возникших при решении данной задачи;
- у новых участников нет возможности ознакомиться с деталями обсуждений.

Работая в АС учета задач, у всех участников есть возможность обсудить все вопросы прямо в контексте конкретной задачи. При этом история обсуждения, т.е. каждое сообщение, будет доступно для чтения всем заинтересованным участникам, как текущим, так и тем, которые подключаться к задаче в будущем, для выполнения или анализа данной задачи.

Существуют большое множество задач, связанных с обеспечением процесса разработки интернет-проектов, которые могут решаться проще и удобнее, и следовательно, быстрее, если использовать для этих целей АС для учета задач при разработке интернет-проектов.

Актуальность обзора. Чтобы рассмотреть вопрос актуальности обзора различных АС, следует коснуться задач, которые решаются с помощью них – АС для учета задач при разработке интернет-проектов упрощает обработку, хранение и передачу различной информации между участниками системы.

Сталкиваясь впервые с необходимостью внедрения подобной АС в бизнес-процессы организации, нужно понимать, какие именно задачи должна решать эта система, а также понимать, чем отличаются между собой наиболее популярные системы для ведения учета задач при разработке интернет-проектов.

Чтобы достичь наибольшей эффективности при работе в АС, следует использовать одну систему для всех задач. Относительно большое количество информационных систем подобного рода требует детального изучения различий между ними, чтобы сделать объективный выбор той системы, которая решает максимальное количество задач организации и подходит по всем критериям.

Любая компания, занимающаяся разработкой интернет-проектов имеет схожие процессы разработки, тем не менее, в каждой компании они все равно уникальны. И набор требований к АС тоже будет отличаться.

Поиск АС для сравнения, выявление основных параметров для сравнения, и проведение самого анализа занимает много времени. При этом объективные результаты сравнения всех систем (без учета изменений в самих системах, за период между сравнениями) всегда будут одинаковыми, сколько бы много различных людей этим не занималось.

Результатами сравнительного обзора смогут воспользоваться в своих целях различные представители организаций по разработке интернет-проектов, которые задались целью поиска АС учета задач для внедрения на предприятии и оптимизации бизнес-процессов компании.

Таким образом, можно сказать, что сводная таблица, содержащая в себе результаты сравнения всех функциональных возможностей различных систем автоматизированного учета задач, является актуальным предметом для исследования и проведения сравнительного обзора существующих АС.

Предмет сравнения. Предметом сравнения являются различные автоматизированные системы для ведения учета задач при разработке интернет-проектов.

Каждая АС обладает уникальным набором функциональных возможностей, и как следствие этого – разные АС подходят разным компаниям, в зависимости от их требований к АС.

Чтобы провести сравнение АС, следует выделить параметры для их сравнения, на основании наиболее актуальных требований к АС данного рода.

Основные возможности АС для сравнения:

- создание пользователей с различными уровнями прав (ролями);
- создание задач;
- группировка задач;
- обсуждение задач (комментирование);
- возможность прикладывать материалы;
- назначение ответственных;
- указание сроков выполнения задачи;
- учет затраченного времени;
- возможность распределять приоритеты (порядок выполнения задач);
- напоминания ответственным о задачах;
- уведомления о новых событиях и изменениях, связанных с задачами;
- формирование различных отчетов;
- отображение планового графика выполнения задач (диаграмма Ганта);
- быстрый поиск нужной информации;
- наличие веб-версии АС;
- наличие мобильной-версии АС (приложение для телефона);
- стоимость лицензии/подписки.

Перечень АС для сравнения. Возьмем для сравнения наиболее распространенные автоматизированные системы для учета задач при разработке интернет-проектов.

1. Jira – очень гибкий веб-сервис, с помощью которого команды разработчиков могут планировать проекты, назначать ответственных у задач, планировать спринты и собирать задачи в общий список, выставлять последовательность и крайние сроки.
2. Slack – онлайн-сервис (есть различные приложения для десктопа, iOS и Android) для создания командных чатов, где можно вести обсуждения с командой и клиентами.
3. GanttPro – диаграмма Ганта, сервис позволяет планировать и управлять проектами через веб-интерфейс, графически представлять процессы, создавать задачи и привязывать их к участникам, выставлять сроки и процент завершения частных задач и проекта в целом, а также делиться полученным графиком Ганта с командой и с клиентами с различными правами (просмотр или редактирование).
4. Asana – веб-сервис и мобильное приложение для командной работы без использования традиционной электронной почты. Предлагает как бесплатное использование (в командах до 14 сотрудников), так и платный пакет услуг для больших команд. Отличается простым и универсальным интерфейсом – в нем могут вести свои проекты как разработчики, так и дизайнеры.
5. Basecamp – один из самых популярный онлайн-сервисов для совместной работы над проектами, позволяющий делиться документами, вести обсуждения с командой, создавать

to-do листы и добавлять комментарии к задачам, высылать и принимать электронную почту.

6. Trello – простой и удобный инструмент, позволяющий упорядочить задачи, обсуждения и идеи на одной доске, весь проект «как на ладони», видны сразу все задачи по проекту, кто над чем работает и многое другое.
7. Bitrix24 – АС, включающая в себя учет задач, планирование проектов и времени их выполнения, графическое представление всей информации на календаре, есть возможность создавать виртуальные рабочие группы и получать уведомления.
8. Мегплан – система для совместной работы в малой или средней компании любого профиля. Помогает контролировать работу удаленных сотрудников. Среди основных возможностей: учет задач, файловый сервер, внутренняя почта, встроенный чат и др.

Сравнение АС. Параметров для сравнения получается достаточно много, их трудно поместить в одну, удобно читаемую таблицу. Потому сформируем таблицу для сравнения по наиболее важным критериям среди всех выбранных параметров.

Таблица. Сравнение по наиболее важным критериям среди выбранных параметров

АС	Критерии					
	Создание задач	Обсуждение задач	Назначение ответственных	Учет времени	диаграмма Ганта	Стоимость
Jira	+	+	+ (только один чел.)	+	+	от 10\$/мес
Slack	–	+	–	–	–	0, от 7\$/мес
GanttPro	+	+	+	+	+	0, от 19\$/мес
Asana	+	+	+	+	–	от 8,5\$/мес
Basecamp	+	+	+	+	+	от 99\$/мес
Trello	+	+	+	–	–	0, от 10\$/мес
Bitrix24	+	+	+	+	+	0, от 990 руб/мес
Мегплан	+	+	+	+	+	от 330 руб/мес

На основании полученной таблицы можно сделать вывод, что все самые популярные АС обладают наиболее ожидаемым от них функционалом: учет всей связанной информации по проектам, задачам и исполнителям, а также обсуждение задач.

Все АС предполагают работу с веб-сервисом, через сайт или мобильное приложение (нет в таблице).

Наибольшие различия наблюдаются в ценовой политике, оплата предполагается за пользователя системы в месяц. Все АС имеют либо бесплатный пробный период, либо бесплатный режим с ограниченными возможностями. Таким образом, у пользователей есть возможность детально познакомиться с системой бесплатно.

Заключение. В данном сравнительном обзоре рассмотрены наиболее популярные автоматизированные системы для учета задач при разработке интернет-проектов и наиболее важные функциональные возможности, которыми должны обладать подобные системы.

На текущий момент существует очень много АС для учета задач по проектам. Все они немного отличаются своими возможностями и сильно отличаются внешним видом (дизайном), однако каждая из рассмотренных систем обладают всеми необходимыми, в первую очередь, возможностями.

Данный обзор должен помочь внести ясность, на что следует обратить внимание при выборе АС, и помочь сделать выбор наиболее подходящей АС для потребностей читателя.

Литература

1. Богданов В. Управление проектами. – М.: МИФ, 2012. – 248 с.
2. Сазерленд Д. Революционный метод управления проектами. – М.: МИФ, 2015. – 288 с.
3. Обзор сервисов для управления проектами [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/276873/>, своб.



Дмитриевская Алёна Александровна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: dmitalyona@yandex.ru



Мустафина Ляйсан Ирековна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4108

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: amix93@mail.ru

УДК 004.4

ВЕБ-СРЕДА КАК ОДИН ИЗ ВАРИАНТОВ ПРИМЕНЕНИЯ КАРТОГРАФИЧЕСКИХ ИНТЕРФЕЙСОВ

Дмитриевская А.А., Мустафина Л.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 наиболее часто используемые картографические интерфейсы, области их применения, основные инструменты, а также ключевые особенности.

Ключевые слова: картографический интерфейс, API, карта, веб-картография, веб-среда.

На сегодняшний день существует множество вариантов размещения карт в веб-приложениях. В статье «Использование API веб-картографических сервисов для доступа к геоданным» Г.В. Потапов и М.Ю. Потанин рассматривают Google Maps API, реализованные на JavaScript, в качестве одного из возможных картографических интерфейсов реализации доступа к геоданным [1]. В работе «Веб-гис технологии: обзор основных направлений развития» М.Ю. Потанин уделяет внимание вопросу областей

применения, особенностей и преимуществ веб-гис технологий [2]. Анализ исследований, посвященных проблеме картографических интерфейсов, показал, что вопросы выбору того или иного картографического интерфейса остались вне поля зрения ученых [1, 2].

В настоящее время существует несколько наиболее часто употребляемых картографических API-сервисов для размещения карт в веб-среде. Среди них наиболее широкое распространение получили:

- Google API;
- Яндекс API;
- 2GIS API.

Google API-карт – самый распространенный многофункциональный картографический интерфейс в мире. Позволяет разместить на своем веб-сервисе как автоматически сформированную карту, так и с пользовательскими настройками. Функционал основных инструментов карт Google представлен в табл. 1.

Таблица 1. Основной набор инструментов интеграции карт Google maps [3]

Наименование приложения	Функционал
Google Maps Geocoding API	позволяет преобразовывать адреса и географические координаты
Google Places API Web Service	позволяет реализовать автозаполнение и добавлять актуальную информацию о различных местах на веб-сервис
Google Maps Elevation API	позволяет получить данные о высоте для любой точки мира
Google Maps Distance Matrix API	позволяет рассчитать время в пути и расстояния между разными точками
Google Maps Roads API	позволяет активировать функцию переключения на дорогу для точного отслеживания пути по GPS
Google Maps Time Zone API	позволяет получить информацию о часовом поясе для любой точки мира
Google Maps Geolocation API	позволяет осуществить поиск местоположения на основе информации, передаваемой по сотовой связи и Wi-Fi
Google Maps Direction API	позволяет рассчитать маршрут между различными точками

Основные особенности карт Google API:

- возможность детального просмотра улиц;
- предоставление гиперссылок на имеющиеся рейсы/билеты на самолет/в кинотеатр просматриваемой страны/города;
- возможность планирования маршрута, состоящего из нескольких точек;
- возможность сохранения интересующего адреса;
- наличие упрощенного режима при нестабильном подключении к сети Интернет;
- возможность измерить расстояние между объектами;
- использование карт Google в качестве навигатора по местности/помещению.

Яндекс API-карт – один из самых популярных картографических сервисов, используемых в России. Функционал основных инструментов карт Яндекс.Карт представлен в табл. 2.

Особенности API Яндекс.Карт:

- использование карт Яндекс в качестве навигатора в пути;
- возможность построения маршрута в условиях загруженности дорог.

Таблица 1. Основной набор инструментов интеграции Яндекс.Карт [4]

Наименование приложения	Функционал
JavaScript API	набор компонентов для размещения интерактивных Яндекс.Карт на страницах сайта или в веб-приложении
Платный API Карт	позволяет приобрести коммерческую лицензию для закрытых систем и мониторинга транспорта
Геокодер	предоставляет API для перевода географических координат в адрес и наоборот
Static API Карт	позволяет разместить статичные изображения Яндекс.Карт в веб-сервисе, отдаваемые по HTTP-запросу
API Поиска по организациям	программный интерфейс, позволяющий осуществлять поиск организаций и топонимов
YMapsML	предоставляет XML-язык описания географических данных, разработанный в Яндексе
API Конструктора карт	предоставляет интерфейс для настройки параметров созданной в Конструкторе схемы

2GIS API-карт – один из наиболее известных веб-сервисов карт, позволяющих их эксплуатацию без подключения к сети Интернет. Функционал основных инструментов карт 2ГИС представлен в табл. 3.

Таблица 2. Основной набор инструментов интеграции 2ГИС карт [5]

Наименование приложения	Функционал
Slot	JavaScript-фреймворк для создания модульных изоморфных приложений
RiftJS	Фреймворк для создания сложных интерактивных web-приложений
MakeUP	Инструмент для контроля за качеством верстки на веб-проектах
Tars	Автосборка статичных (и не только) веб-проектов
Winium	Фреймворк для автоматизации тестирования Win-приложений

API-карт 2GIS имеет большой набор инструментов для разработчиков, открытых в свободном доступе. Отличительными особенностями картографических API 2GIS являются:

- простота использования;
- доступность исходного кода;
- возможность просматривать на карте детальные планы этажей зданий;
- возможность использования карт без подключения к сети Интернет;
- возможность указания входа в организацию.

Проведенные в этом направлении исследования показали, что гибкие настройки и персонализация картографических API позволяют произвести истинную и конкретную настройку под каждого пользователя.

Однако имеет место противоречие между возросшими требованиями к эффективности функционирования API-карт и недостаточными возможностями существующих методических и инструментальных средств по обеспечению указанных требований.

Исходя из этого, возникает необходимость в поиске, разработке и совершенствовании методических и инструментальных средств автоматизации процесса внедрения картографических веб-интерфейсов, обеспечивающих повышение качества представляемых геоданных. Следовательно, проведение исследований на рассматриваемую тему с привлечением методов структурно-функционального анализа, теории системного анализа является актуальным направлением. Проведенный анализ применения картографических API позволит создать рекомендации по выбору API-карт в веб-среде.

Литература

1. Потапов Г.В., Потанин М.Ю. Использование API веб-картографических сервисов для доступа к геоданным // Земля из космоса: наиболее эффективные решения. – 2009. – № 3. – С. 7–13.
2. Потанин М.Ю. Веб-гис технологии: обзор основных направлений развития // Системный анализ в науке и образовании. – 2014. – № 2. – С. 43–52.
3. Google Maps Web Service API | Google Developers [Электронный ресурс]. – Режим доступа: <https://developers.google.com/maps/web-services/?hl=ru>, своб.
4. API карт – Технологии Яндекса [Электронный ресурс]. – Режим доступа: <https://tech.yandex.ru/maps/>, своб.
5. 2ГИС Технологии [Электронный ресурс]. – Режим доступа: <http://techno.2gis.ru/opensource>, своб.



Довженко Михаил Игоревич

Год рождения: 1995

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: michael.dovzhenko@gmail.com



Готская Ирина Борисовна

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, д.педагог.н.,
профессор

e-mail: iringot@mail.ru

УДК 004

АНАЛИЗ API УПРАВЛЕНИЯ BLUETOOTH-УСТРОЙСТВАМИ НА РАЗЛИЧНЫХ ПЛАТФОРМАХ

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 представлена сравнительная характеристика API для управления Bluetooth-устройствами трех наиболее распространенных на мобильных устройствах платформ: Android, iOS и Web.

Ключевые слова: Bluetooth, API, Android, iOS, Web.

Сегодня в мире используется более 10 млрд Bluetooth-устройств и роль Bluetooth-технологии в области беспроводной передачи данных продолжает расти. По оценкам ABI Research [1] к 2021 году по всему миру будут применяться около 48 млрд разнообразных устройств с возможностью обмена данными через Интернет. Из них почти треть будет поддерживать беспроводную технологию Bluetooth. В связи с актуальностью данной технологии в настоящей работе ставилась задача представить различные модели управления устройствами Bluetooth в сопоставлении их возможностей и перспектив развития.

Среди наиболее распространенных и лидирующих платформ, предоставляющих нативную поддержку Bluetooth, можно выделить iOS и Android. Также в 2016 году утверждена первая спецификация Web Bluetooth, которая позволяет веб-сайтам взаимодействовать с Bluetooth-устройствами безопасным способом. Эта спецификация уже реализована в некоторых браузерах, что делает целесообразным анализ и этой многоцелевой платформы.

На основе официальной документации для Android [2], iOS [3] и спецификации Web Bluetooth [4] в таблице приведена сравнительная характеристика Bluetooth API данных платформ.

Языки программирования и средства:

- Android: Разработка приложений с помощью различных Android API-интерфейсов осуществляется в среде языка Java;
- iOS: Разработка приложений с помощью API Apple-платформ (iOS, OS X) осуществляется на основе Objective-C или Swift;
- Web: Web Bluetooth API на данный момент реализован в последних версиях браузеров Google Chrome и Opera и ориентирован на JavaScript.

Объектно-ориентированность:

- Android: В Android API для любых операций с Bluetooth определен объект BluetoothAdapter, который олицетворяет собственный Bluetooth адаптер Android-устройства;
- iOS: Разработка приложений с помощью API Apple-платформ (iOS, OS X) осуществляется на основе Objective-C или Swift;
- Web: В Web Bluetooth API всю Bluetooth-функциональность реализует объект navigator.bluetooth.

Асинхронность и событийная ориентированность:

- Android: Android Bluetooth API предоставляет асинхронное взаимодействие с Bluetooth Low Energy устройствами посредством функций обратных вызовов (callback). Каждый метод предоставляемого API принимает в качестве последнего параметра объект обратного вызова, который, в свою очередь, содержит функции обратного вызова. Эти функции используются для получения результатов клиентских операций;
- iOS: Каждая операция в Core Bluetooth framework является асинхронной. Асинхронность реализуется с помощью шаблона проектирования «delegate-protocol» и callback-методов. Core Bluetooth framework определяет интерфейсы, которые необходимо реализовать. Таким образом, интерфейс CBCentralManagerDelegate определяет ряд callback-методов, которые позволяют отслеживать состояние подключения и сканирования списка устройств. А после подключения к устройству, интерфейс CBPeripheralDelegate определяет все остальные callback-методы непосредственно для взаимодействия с данными удаленного устройства;
- Web: Web Bluetooth API предполагает только асинхронное взаимодействие. Данный интерфейс был разработан с учетом современных подходов к написанию асинхронного кода с использованием паттерна Promise.

Методы и степень их абстракции:

- Android и iOS Bluetooth API абстрагируют низкоуровневые детали из спецификации Bluetooth 4.0. Они предоставляют высокоуровневые методы для выполнения всех типичных Bluetooth Low Energy задач, как для клиентской роли (сканирование удаленных устройств, соединение с ними, взаимодействие с их GATT-сервисами и характеристиками и подписка на уведомления об изменениях значений характеристик), так и для серверной роли (определение сервисов и характеристик на локальном устройстве, трансляция сервисов, ответ на запросы чтения и записи от подключенного central устройства, отправка уведомлений);
- Web Bluetooth API предоставляет аналогичные высокоуровневые методы для реализации типичных, для клиентских приложений, задач.

Модель безопасности. Избегание вредоносного влияния до того как стало известно, что сайт или приложение является вредоносным:

- у нативных (iOS и Android) мобильных приложений хранилища (AppStore, PlayStore) имеют полный доступ к коду приложения и могут тестировать его на предмет безопасности. Однако из-за того, что удаленные Bluetooth-устройства могут разговаривать на различных протоколах, хранилища не могут в полной мере протестировать их и вынуждены разрешать все сообщения, информации о вредоносности которых нет;
- Web: у нас нет возможности делать оффлайн-сканирование веб-сайтов. Однако можно блокировать известные вредоносные сообщения используя обновленный реестр вредоносных сервисов.

Избегание вредоносного влияния после того как стало известно, что сайт или приложение является вредоносным:

- можно удалить все нативные приложения из хранилища, загруженные под одной кредитной картой;
- сервис «Безопасный просмотр» позволяет выявлять и блокировать небезопасные сайты и уведомлять об этом пользователей и веб-мастеров.

Отдельно стоит рассмотреть критерий доступности, т.е. какие протоколы и роли Bluetooth поддерживаются на этих платформах. Это важно, потому что протоколы не совместимы друг с другом. В Bluetooth-технологии существует две роли: клиентская и серверная. Однако для разных протоколов они называются по-разному. В Bluetooth Classic они называются client и server, а в Bluetooth Low Energy – central и peripheral соответственно. Чтобы установить Bluetooth-соединение, необходимо иметь обе. Одно устройство берет одну роль, второе – другую. Роль сервера предназначена для устройства, которое имеет данные, необходимые устройству, выступающему в роли клиента. В таблице представлены протоколы и роли, поддерживаемые различными платформами.

Таблица. Совместимость протоколов, ролей и платформ

Протокол	Роль	
Classic	Server	Client
	– iOS-устройства не могут выступать в роли сервера. Android-устройства могут принимать обе роли.	– iOS-устройства являются клиентами. Android-устройства могут принимать обе роли.
Low Energy	Peripheral	Central
	– iOS-устройства могут принимать обе роли начиная с 6 версии. Android-устройства не способны выступать в роли «peripheral» до версии 4.4	– Web-платформа поддерживает только «central» роль. – iOS-устройства полностью поддерживают обе роли. – Android-устройства могут быть только в роли «central» до версии 4.4

Таким образом, все рассмотренные платформы предоставляют высокоуровневые асинхронные интерфейсы для взаимодействия с Bluetooth-устройствами, но с рядом особенностей. Платформы iOS и Android позволяют своим приложениям выполнять как клиентскую так и серверную роль. В то время как веб-платформа ограничена только «central» ролью, что говорит о ее нацеленности исключительно на интернет вещей. Также, нативные iOS и Android приложения имеют по умолчанию больше возможностей, чем веб-приложения. Например, нативные приложения имеют прямой доступ к сети, могут выполняться в фоновом режиме и взаимодействовать с несколькими удаленными устройствами одновременно. Android- и iOS-платформы имеют устойчивые и хорошо документированные Bluetooth-интерфейсы, в то время как Web Bluetooth-спецификация неустойчива и еще находится на стадии активного развития. С

точки зрения безопасности, перед тем как дать доступ к удаленному устройству, Web Bluetooth предоставляет больше предупреждений для пользователей чем iOS или Android. Веб-интерфейс имеет свои уязвимости, но, тем не менее, вероятность получить несанкционированный доступ к устройству через него значительно меньше. В настоящий момент приложения от производителей умных Bluetooth-устройств зачастую выпускаются только под две платформы – iOS и Android, однако Web Bluetooth API стремительно развивается и имеет большие перспективы, так как позволит разработчикам создавать единое решение, которое будет работать на всех платформах, как мобильных так и десктопных, что означает меньшую стоимость разработки, больше open source библиотек для различных физических продуктов и больше инноваций [5].

Литература

1. The official Bluetooth Membership. Bluetooth Special Interest Group [Электронный ресурс]. – Режим доступа: <https://www.bluetooth.com/specifications/bluetooth-core-specification/bluetooth5>, своб.
2. Android Bluetooth Guide [Электронный ресурс]. – Режим доступа: <https://developer.android.com/guide/topics/connectivity/bluetooth.html>, своб.
3. Apple Core Bluetooth Framework [Электронный ресурс]. – Режим доступа: <https://developer.apple.com/bluetooth>, своб.
4. Web Bluetooth Specification [Электронный ресурс]. – Режим доступа: <https://webbluetoothcg.github.io/web-bluetooth>, своб.
5. The Web Bluetooth Security Model [Электронный ресурс]. – Режим доступа: <https://medium.com/@jyasskin/the-web-bluetooth-security-model-666b4e7eed2>, своб.



Дорохина Оксана Юрьевна

Год рождения: 1986

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4106

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: ksu_sunrise@mail.ru



Шуклин Дмитрий Анатольевич

Год рождения: 1979

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

e-mail: do@limtu.ru

УДК 004.512

ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ ДИЗАЙНА WEB-ИНТЕРФЕЙСОВ С УЧЕТОМ ЦВЕТОВОГО И ЭМОЦИОНАЛЬНОГО ВОСПРИЯТИЯ

Дорохина О.Ю., Д.А. Шуклин

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 понятие эмоциональный дизайн, способы воздействия на эмоциональное восприятие, виды эмоций, влияние цвета на восприятие, критерии юзабилити, составляющие качественного web-дизайна, поэтапные шаги проектирования с учетом цветового и эмоционального восприятия, развитие новых технологий в web-дизайне.

Ключевые слова: эмоциональный дизайн, цветовое восприятие, web-дизайн, проектирование web-интерфейсов, юзабилити.

Эмоциональный web-дизайн основывается на психологическом восприятии пользователем и творческом подходе. Его главной задачей является вызвать нужные эмоции, создать новые впечатления, сделать интерфейс максимально комфортным и ощущение, что с ним взаимодействует человек, а не робот. Правильно спроектированный дизайн будет способствовать положительным отзывам, этого можно достигнуть благодаря дизайнерским элементам: изображениям эмоций, фразам, фотографиям, анимации, цветам и т.п. Люди, не осознавая будут рекламировать сайт знакомым и друзьям. Это и есть главная цель эмоционального дизайна.

Эмоции бывают различных видов и проявляются в виде настроения, ощущений, чувств и переживаний. Из них выделяют семь базовых: радость, грусть, презрение, страх, отвращение, удивление и гнев. Они выражаются с помощью мимики и жестов и являются для всех универсальными. В этой связи при проектировании можно размещать изображения с одной из универсальных эмоций в зависимости от концепции, при этом важно, чтобы они были искренними, иначе пользователь сразу почувствует фальш. Прежде всего, нужно определить, какие эмоции необходимо вызвать у целевой аудитории, собрать больше информации не только о демографическом составе, но и психографические данные, например, какие эмоции мотивируют или будут мотивировать различные группы людей [1].

Качественный дизайн основывается на четырех обязательных составляющих, создающих психологический комфорт и удовлетворяющих потребностям пользователей при взаимодействии с интерфейсом:

1. функциональность;
2. надежность;
3. удобство;
4. эстетическое удовольствие [2].

Юзабилити – определяет степени эффективности, трудоемкости и удовлетворенности, с которыми web-интерфейс может быть использован пользователями при определенном контексте эксплуатации для достижения необходимых целей. Существуют различные критерии оценки юзабилити интерфейсов веб-сайтов, многие исследователи определяют свои. Основные из них:

- понятность, т.е. страницы сайтов должны быть максимально просты, чтобы в них мог разобраться любой человек;
- скорость работы пользователя. При проектировании качественного сайта, следует уделить внимание скорости, с которой пользователь сможет выполнить какую-либо работу;
- субъективное удовлетворение функциональностью и эстетикой web-интерфейса;
- удобная структура, пользователь должен легко ориентироваться в интерфейсе;
- устранение избыточного контента, так как часто на сайтах можно встретить большое количество информации, картинок и т.п. Это только занимает лишнее место и отвлекает внимание от главного, из-за чего пользователь просто теряется;
- количество человеческих ошибок, некачественно созданный интерфейс может повлечь за собой снижение эффективности работы с системой;
- субъективное удовлетворение функциональностью и эстетикой web-интерфейса [3, 4].

Важное значение в web-дизайне играет цветовое восприятие, ведь цвет способен влиять на подсознание и вызывать определенные эмоции. Известно, что теплые цвета ассоциируются с закатом, светом, огнем, обычно они побуждают к действиям, поэтому в некоторых случаях в них удобно окрашивать, например, кнопки. Но не желательно окрашивать в красный, так как он часто ассоциируется со знаком «стоп», хотя всегда бывают исключения. Холодные цвета успокаивают, умиротворяют, придают глубину в них можно окрашивать, например, фон. При подборе цветовой

гаммы важно не переборщить с их количеством, ведь чем больше цветов, тем более странным может получиться дизайн. Большинство сайтов известных компаний используют небольшое количество цветов, примерно менее пяти.

Существуют определенные цветовые ассоциации, влияющие на психологическое восприятие. Но не стоит им следовать строго, тем более, что в сочетании с другими цветами визуальное восприятие может измениться:

- красный цвет: активный, энергичный, страстный и агрессивный. Он пробуждает эмоции, стимулирует к принятию решений. Обычно его используют для привлечения внимания;
- синий является цветом спокойствия, уравновешенности. Также ассоциируется с холодом, лояльностью, интеллектом, точностью и общением. Он распространен в высокотехнологических компаниях. Прекрасно сочетается с различными оттенками серого и зеленого. Если синий используется в качестве основного, то оттенки оранжевого или желтого можно использовать в активных элементах;
- цветом оптимизма, уверенности, веселья и дружелюбия является желтый. Он символизирует счастье, но слишком много желтого, так же как и любого другого цвета, может угнетать. Хорошо сочетается с черным и синим;
- зеленый тонизирует, успокаивает, расслабляет и снижает стресс. Ассоциируется с удачей, дружелюбием, здоровьем и т.п. В сочетании с коричневым успокаивает, а при сочетании с оттенками лайма или моря бодрит;
- оранжевый достаточно теплый цвет. Символизирует дружелюбие, веселье, игривость, безопасность. Он привлекает и побуждает, поэтому в него часто окрашивают кнопки. Замечательно сочетается с бирюзовым цветом, который является базовым, а оранжевый – акцентирующим. Интересное сочетание получается с фиолетовым;
- фиолетовый связан с креативностью, величием, правдивостью, богатством и загадочностью. Необычным и дорогим получается дизайн в сочетании с золотистым и цветом лайма;
- черный ассоциируется с тяжестью, силой, изысканностью, гламуром и авторитетом. Лучше сочетать его со светлыми цветами, так как при сочетании с темными, дизайн может казаться грустным и скучным;
- белый – это чистота, простота, ясность, символизирует свет. Обычно его используют для фонов.

При проектировании удобно руководствоваться поэтапными шагами, в которых учитываются психологические аспекты восприятия:

1. планирование. На этом этапе нужно подробно описать проект компании, цели, требования и пожелания. Они помогут определиться с планом над которым нужно будет поработать перед переходом к этапу проектирования дизайна. Необходимо определиться с важной информацией: кто целевая аудитория, структурировать информацию, создать карту сайта. Этот шаг важен и не стоит его пропускать;
2. создание набросков. Поиск вдохновения, создание набросков и заметок новых идей. Дизайнеры мыслят образно, поэтому зарисовка идей сохранит их для последующих этапов проектирования;
3. типографика. Является очень важным элементом сайта, рекомендуется именно с нее начинать проектирование;
4. цвета. Подбор основного цвета, вторичного, заднего фона и активных цветов для сайта, сохранение цветовой палитры;
5. создание прототипа и макета. Планирование визуализации макетов всех вариантов web-страниц и выбор подходящего;
6. итоговый шаблон дизайна. Объединение всех дизайнерских элементов [5].

Технологии развиваются, каждый год появляются новые тренды. В последнее время пользуются популярностью сайты с эффектом присутствия. Это осуществляется благодаря графике, 3D-фотографиям и т.д. Существуют интернет-магазины, в которых максимально проработан эффект присутствия, чтобы покупатель мог ощутить себя как в реальном магазине, можно

рассматривать продукт со всех ракурсов, увеличивая, уменьшая и вращая, также можно примерить товар. Многие сайты становятся приложениями, в которых можно играть или осуществлять какие-либо действия. В будущем предвидят, что трехмерная web-реальность придет на смену двумерной, станет более реалистичной.

Литература

1. Уэйншенк С. 100 главных принципов дизайна. – СПб.: Питер, 2012. – 272 с.
2. Уолтер А. Эмоциональный дизайн. – Изд-во: Манн, Иванов и Фербер, 2012. – 144 с.
3. Головач В.В. Дизайн пользовательского интерфейса: Искусство мыть слона. – Изд-во: N/A, 2010. – 97 с.
4. Круг С. Веб-дизайн: книга Стива Круга или «Не заставляйте меня думать!» / Пер. с англ. – 2-е изд. – СПб.: Символ-Плюс, 2008. – 224 с.
5. Rafal T. The Essential Web Design Handbook. – Изд-во: RafalTomal.com, 2015. – 210 p.



Дубро Андрей Сергеевич

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: dadron13@yandex.ru

УДК 004.4`2

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ ДЛЯ КРОССПЛАТФОРМЕННОЙ РАЗРАБОТКИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Дубро А.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 работе произведено сравнение кроссплатформенных инструментальных средств для разработки мобильных приложений.

Ключевые слова: кроссплатформенная разработка, Cordova, Xamarin, Appcelerator Titanium, Telerik Platform, Kony Platform, IBM Worklight.

В настоящий момент в мире существует большое количество различных платформ, самые крупные из которых имеют свои магазины приложений. Apple и Microsoft для написания функционала к их операционным системам используют Swift и C#. Оба языка имеют хорошо спонсируемую базу для своих платформ. Однако какое бы ни было название у языка – суть работы не меняется. Людям все также будут нужны мобильные приложения, основой которых будет оставаться логика, производительность, память и возможность взаимодействовать с другими приложениями и людьми. Ускорить процесс разработки и использовать весь необходимый большинству пользователей функционал позволяют технологии кроссплатформенного создания мобильных и десктопных приложений [1–3].

Кроссплатформенные приложения – это приложения, которые создаются сразу же под несколько платформ. Основным их преимуществом является возможность использования одного и того же кода для написания приложений под разные операционные системы, что существенно сокращает время разработки.

Нативные приложения – это приложения, создающиеся под конкретную операционную систему, использующие ее технические возможности и включают все дизайнерские и интуитивные особенности.

Одними из самых популярных инструментов для кроссплатформенной разработки мобильных приложений являются:

- Appcelerator Titanium;
- Kony Platform;
- Cordova;
- Telerik Platform;
- IBM Worklight;
- Phonegap;
- Xamarin.

Appcelerator Titanium – программное обеспечение, основанное на Eclipse. Основное преимущество – возможность разрабатывать нативные приложения, используя JavaScript, PHP, Ruby, Python. Возможно покупать необходимые компоненты, обеспечивающие большой функционал и удобства для пользователя. Есть встроенные коннекторы для связи с самыми популярными приложениями. Из недостатков самого продукта можно отметить медленную компиляцию и недостаточно подробную документацию.

Работа с Kony Platform происходит в собственной среде на базе Eclipse – Kony Studio. Kony Platform позволяет одновременно разрабатывать приложения для телефонов, планшетов, десктопных устройств и браузеров при помощи одного языка программирования. Поддержка всего цикла создания приложения: дизайн, разработка, тестирование, развертывание, управление кроссплатформенными приложениями. Большим плюсом является удобная совместная работа разработчиков и дизайнеров, так как есть возможность делиться прототипами и приложениями. Имеется система отчетов и логирования. Программное обеспечение платное, цена составляет 19\$ в месяц.

Xamarin позволяет создавать нативные приложения на языке C#. Приложения имеют максимальную производительность, благодаря тому, что на стадии выполнения код сразу же компилируется в нативный. Имеется сервис виртуального тестирования практически на любых устройствах, называемый Xamarin Test Cloud. Присутствует система встроенных покупок в приложении. Также есть множество плагинов, увеличивающих возможности Xamarin. Работать с Xamarin можно как в их собственной IDE Xamarin Studio, так и в привычной для многих Visual Studio при наличии соответствующего плагина. Стоимость 99\$ за платформу. Бесплатно испытать фреймворк можно в течение 3-х месяцев.

Cordova является инструментом с открытым кодом и на ее основе созданы другие фреймворки, такие как: Telerik Platform, Phonegap, IBM Worklight. Позволяет писать приложения используя HTML, CSS, JS, поэтому веб-разработчикам намного проще начать создавать свои программы. Доступ к нативной функциональности устройств осуществляется за счет плагинов, которые, в свою очередь, замедляют работу приложения, а некоторые из них работают не корректно.

У Telerik Platform отличия от Cordova следующие: имеется система контроля версий, возможно создавать дизайн и прототипы, имеет собственный портал, с помощью которого предоставлять доступ к приложению по своему выбору. Для написания приложений используются технологии .NET, Java, JS, PHP.

Phonegap разработано фирмой Adobe. Готовые приложения функционируют как веб-страница, поэтому приложения создаются при помощи HTML, CSS и JavaScript. Главное отличие от Cordova – наличие облачного сервиса. Однако в Phonegap присутствуют баги, которых нет при использовании Cordova.

IBM Worklight так же как и в PhoneGap можно создавать web-приложения, и дополнительно гибридные приложения. Присутствует аналитика, возможность интернационализации,

безопасность на каждом уровне приложения. Используются языки: Java, Objective-C, C# для нативных и гибридных приложений, JavaScript для web.

Ниже в таблице приведен список операционных систем и технологий, под которые возможна разработка с помощью определенного инструмента.

Таблица. Список операционных систем и технологий

ПО	IOS	Android	BlackBerry	Windows Phone	Web Brouser	Mobile Web
Appcelerator Titanium	+	+	+	–	–	+
Kony Platform	+	+	+	+	+	–
Xamarin	+	+	–	+	–	–
Cordova	+	+	+	+	–	+
Telerik Platform	+	+	–	+	–	–
Phonegap	+	+	+	+	–	+
IBM Worklight	+	+	+	+	–	+

Из представленных сведений можно сделать выводы об инструментах, понять, какие возможности предоставляет каждый из них и что необходимо лично Вам.

Для начинающих разработчиков важную роль играет порог входа в разработку, стоимость продукта и наличие подробной документации. Под данные критерии лучше всего подходит Cordova и все образованные от нее фреймворки, а также Appcelerator Titanium при предпочтении на нативной составляющей.

Команде специалистов сильно ускорит процесс разработки работа с Kony Platform. Сочетание качественного дизайна и функциональности играет большую роль на рынке мобильных приложений.

Литература

1. Обзор самых популярных кроссплатформенных мобильных фреймворков [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/229559/>, своб.
2. Appcelerator – отзывы с фронта [Электронный ресурс]. – Режим доступа: <http://alexsorokin.ru/2011/10/appcelerator-issues/>, своб.
3. Введение в Cordova: Основы / Apache Cordova [Электронный ресурс]. – Режим доступа: <http://siteacademy.ru/html/an-introduction-to-cordova-basics>, своб.



Жигальцова Елена

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4206

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: zhigaltsova.e@gmail.com

УДК 004.921

ОСОБЕННОСТИ ИНФОГРАФИКИ В СЕТИ ИНТЕРНЕТ

Жигальцова Е.

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 представлены результаты изучения инфографики в сети Интернет. Рассмотрены основные виды инфографики в сети и их особенности от оффлайновой инфографики. В работе обоснована актуальность исследования данного направления, предложена классификация областей применения инфографики в сети, рассмотрены актуальные способы создания инфографики.

Ключевые слова: инфографика, инфографика в Интернете, основные виды, особенности, области применения, способы создания.

В современном мире инфографика определяется как «визуализация данных и идей». Использование инфографики уместно во всех областях визуального распространения информации. В XXI веке больший процент среди областей занимает сеть Интернет. По способу отображения информации инфографику в Интернете можно подразделить на следующие виды: статичная, динамическая (анимация, видеоролики) и интерактивная [1–5].

Статичная инфографика зародилась задолго до появления Интернета. Первыми примерами инфографики можно считать исторические карты, созданные более 4000 лет назад. В современном мире статичная инфографика в Интернете представлена как отдельными маленькими блоками, так и большими по размеру и содержанию элементами. Особенностью статичной инфографики в интернете по сравнению с инфографикой для печатных СМИ является возможность сделать его очень большой. Размер инфографики в печатном издании ограничен размерами издания, инфографика, размещенная на сайте, может быть довольно большой, с множеством разделов, информацию из которых можно получить, листая страницу вниз.

Динамическая инфографика – инфографика с анимированными элементами. Основными видами динамической инфографики в интернете являются видеоинфографика, анимированные изображения, презентации.

Интерактивная инфографика подразумевает взаимодействие пользователя с системой отображения информации. Она используется в электронных системах различного рода и на носителях информации: гаджетах, терминалах, сайтах, десктопных и мобильных приложениях.

Первая система интерактивной визуализации данных была создана в 1974 году. С появлением сети Интернет информационная среда расширяется и усложняется, увеличивается потребность в новом уровне взаимодействия с информацией. Интерактивная визуализация позволяет не только изучать информацию в графическом виде, но и влиять на ее структуру с возможностью получения ответной реакции.

Выбор направления инфографики зависит от области распространения инфографики в сети Интернет и от целей создания инфографики.

Инфографика активно используется в самых разнообразных областях от статистики и науки до образования и рекламы. И если вне Интернета преобладает статичная (в газетах, книгах, бизнес-отчетах, инструкциях) и динамическая (телевидение, презентации), то в сети Интернет очень распространена интерактивная инфографика.

В зависимости от цели донесения информационного материала существует несколько направлений использования инфографики в сети.

1. Образование. Основной целью инфографики является информирование. Уровень усвоения научного знания зависит от доступности, восприятия, понимания объяснения материала. Инфографика совмещает логику построения, яркость образов, исчерпываемость информации и рассматривается как новое средство визуализации учебного знания. Инфографические элементы используются не только в дистанционном образовании, но и в научно-популярных целях. Инфографика в образовании – это размещение всей важной информации из выбранной области в одном месте и ее удобное, структурированное изложение.
2. Реклама. Благодаря оригинальному и уникальному образу инфографика позволяет привлечь внимание потребителей. В Интернете инфографика широко применяется для демонстрации товаров и услуг. Наиболее популярным движущим рычагом в продвижении товаров и услуг

в Интернете на сегодняшний день является SMM (Social Media Marketing), который использует инфографику в социальных сетях, блогах и форумах в качестве основного инструмента. Инфографика в рекламе используется в баннерах и плакатах, в рекламных видеороликах. Грамотно составленная инфографика обладает вирусным эффектом.

3. Повествование. Задачей инфографики является не столько передача статистики, сколько помощь пользователю в оценивании ситуации, наглядное сравнение нескольких ситуаций. Инфографика работает там, где нужно показать устройство и алгоритм работы, соотношение предметов и фактов во времени и пространстве, продемонстрировать тенденцию, показать, как что-либо устроено, организовать большие объемы информации.

Аналитическая инфографика отражает статистические и цифровые данные. Сравнение изменений, повествование хронологии событий, истории успеха, описание преимуществ – эти свойства инфографики актуальны в новостных лентах и бизнесе.

Визуализация процесса – даже такие сложные процессы как организация безопасности облачного хранения выглядит доступно в инфографическом стиле повествования. Множественные вариации на тему, как что-то сделать легко и быстро, сравнение разного рода явлений и процессов, представление достижений как компании, так и личных (портфолио). Повествование является крупной сферой применения инфографики в Интернете.

Создание инфографики для сети Интернет. Визуально инфографику представляют в разнообразных формах. Ими могут быть простые рисунки, эмблемы, иллюстрации, диаграммы или карикатуры. Выбранный образ подходит, если он показывает эффективность при передаче информации для выполнения поставленных целей составителем инфографики.

Для создания статичной инфографики используются графические редакторы. Наиболее популярными являются Adobe Illustrator, Adobe Photoshop, Adobe InDesign. Качественную инфографику позволяют сделать и PowerPoint и Paint, при использовании которых не требуются определенные навыки. Создать отдельные инфографические элементы можно в Excel.

В сети существует большое множество сайтов для создания статичной инфографики. Большинство из них англоязычные и платные, но использование таких сайтов позволяет легко внедрить полученную инфографику на сайте или блоге с возможностью адаптации размеров инфографики под сайт.

Для создания динамической инфографики для сайтов используется Adobe Photoshop. Внедрение на сайт происходит либо фиксированной анимацией, либо при помощи языков разработки web-сайта. Наиболее популярными средствами создания видеоинфографики являются Adobe After Effects и Adobe Flash.

Отрисовка отдельных элементов интерактивной инфографики осуществляется в программах Adobe Illustrator и Adobe Photoshop. Добавления функционала инфографики осуществляется посредством программирования. Добавление интерактивности инфографики в Интернете происходит посредством применения технологий HTML5, CSS3 и JavaScript.

По данным статистики We are social в России более 100 млн пользователей интернета и около 70 млн аккаунтов в социальных сетях, значит, направленность инфографики на социальные сети является наиболее эффективной для ее распространения в сети Интернет. По данным TNS Web Index наиболее популярными устройствами для выхода в Интернет на 2016 год являются ПК и ноутбуки – 65%, смартфоны – 54%. В вопросе тенденции роста аудитории пользователей смартфоны обладают 15% увеличения количества пользователей, тогда как количество пользователей ПК и ноутбуков для выхода в Интернет сократилось на 2% за 2016 год. Опираясь на статистические данные, настоящее время актуальными являются развитие инфографики под мобильные устройства с использованием HTML5, интерактивная инфографика, отслеживающая процессы в реальном времени, социальная аналитика и пользовательский дизайн интерфейсов. В связи с актуальностью и большими перспективами развития, эти направления разработки и применения инфографики требуют дальнейшего тщательного исследования.

Литература

1. Фролова М.А. Визуализация образовательного процесса средствами инфографики // Международный научно-исследовательский журнал. – 2016. – № 4(46). – Ч. 3. – С. 104–106.
2. Осатина А.И., Попова И.С. Инфографика как средство визуальной коммуникации / А.И. Осатина // Электрон. сб. статей по материалам XX Международной научно-практ. конф.: «Научное сообщество студентов XXI столетия». – 2014. – № 5(20). – С. 38–47.
3. Tufte E.R. The Visual Display of Quantitative Information [Электронный ресурс]. – Режим доступа: <http://www.colorado.edu/geography/foote/maps/assign/reading/TufteCoversheet.pdf>, своб.
4. Константинова М.В. Реклама как пример инфографики в обучении иностранному языку // Международный научно-исследовательский журнал. – 2016. – № 2(44). – Ч. 4. – С. 25–26.
5. Шкляр Т.Л., Дунин Д.М. Социальные сети как инструмент маркетинговых коммуникаций // Экономика и современный менеджмент: теория и практика: сб. ст. по матер. XXXIX междунар. науч.-практ. конф. – 2014. – № 7(39).

**Жирнова Марина Анатольевна**

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: marjnazhjrnova@yandex.ru

**Государев Илья Борисович**Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна,
к.педагог.н., доцент

e-mail: ilia.gossoudarev@gmail.com

УДК 004

**ПРИМЕНЕНИЕ UML ДЛЯ ПРОЕКТИРОВАНИЯ И РАЗРАБОТКИ СИСТЕМ
НА ЯЗЫКЕ JAVASCRIPT****Жирнова М.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 описан метод применения UML для проектирования и разработки систем на языке JavaScript стандарта ECMAScript 2015 на примере диаграммы классов. Рассмотрено применение UML в разработке системы на прототипно-ориентированном языке JavaScript, описаны особенности спецификации языка ECMAScript 2015.

Ключевые слова: UML, JavaScript, кодогенерация, проектирование, ХМІ.

При проектировании информационных систем важную роль занимает коммуникация исполнителя с заказчиком. Во время процесса коммуникации необходимо правильно интерпретировать требования и найти оптимальный подход к решению поставленной задачи. Таким образом, первостепенное значение приобретает выбор языка для общения заказчика и исполнителя. Для данных целей все чаще используется UML. Основная особенность данного языка в визуальном представлении компонентов системы.

UML в разработке системы. К целям создания UML-моделей относят не только общение между заказчиком и исполнителем, но и последующее моделирование, документирование и кодирование системы. При проектировании приложения любой сложности важное место занимает разделение компонента на несколько уровней абстракций. Кроме всего прочего, UML-модели разделяют понятия логической структуры и физического хранения данных. На основе UML создается код на языке высокого уровня, а также физическая модель БД. Таким образом, можно сделать вывод, что UML участвует не только в проектировании системы, но и в ее разработке. Для сложных систем применяется обратный процесс кодогенерации – генерация UML-модели на основе существующего кода. Эта функция полезна при реверс-инжиниринге. Кроме всего, UML используется как графическое отображение алгоритма наравне с блок-схемой [1].

Разработка программ на языке JavaScript. С появлением программной платформы Node.js JavaScript стал применяться не только для программирования клиентской стороны веб-приложения, но и для написания программ серверной части веб-приложений, для создания локальных desktop-приложений, а также для программирования микроконтроллеров. Таким образом, был увеличен перечень видов задач, решаемых на языке JavaScript.

Возможность применения UML для проектирования систем на JavaScript. Так как UML создан для объектного моделирования, то он применяется в разработке программ с объектно-ориентированной парадигмой программирования.

Развитие JavaScript предполагает движение в сторону создания объектно-ориентированной среды. В частности, появление в последнем стандарте языка (ECMAScript 2015) слова class, наследование классов с extends, static-свойств и методов get и set. Но на данный момент не реализованы слова для области видимости (private, public, protected), а все методы и свойства доступны вне класса [2].

Таким образом, можно сделать вывод о том, что UML может применяться для разработки программ на языке JavaScript.

Среди рассмотренных [3] редакторов UML-моделей для веб-приложений не было найдено систем, позволяющих автоматически генерировать JavaScript-код. Далее предлагается метод, который позволяет осуществить генерацию на рассмотренный язык.

Генерация JavaScript-кода на основании диаграммы классов UML-модели. Для того чтобы не быть зависимым от конкретного редактора UML, предлагается для генерации JavaScript-кода использовать XMI-стандарт для обмена данными о созданных UML-моделях [4].

В данной работе была рассмотрена генерация диаграммы классов на тестовом примере. Схема работы генератора представлена на рис. 1.



Рис. 1. Схема генерации JS-кода на основании XMI

Рассмотрим создание диаграммы классов пользователя тестового приложения. На диаграмме представлено два класса User и Admin, а также отношение наследования (рис. 2).

В файле формата XMI создается описание графического представления класса в формате XML. Фрагмент файла представлен на рис. 2.

Алгоритм генерирует JS-файл с указанными классами, отношениями и конструктором, где указываются основные атрибуты класса.

Генератор JS-кода реализован на программной платформе Node.js. Для работы с файловой структурой используется встроенный модуль «fs». Для парсинга XMI используется модуль «xml2js». Генератор работает на базе конечных автоматов. Состояниям входных данных из файла XMI ставятся в соответствие синтаксические конструкции JS-языка. Например, состоянию наследования ставится в соответствие слово «extends», происходит переход к состоянию входных данных наименования класса родителя, а также в создании конструктора осуществляется переход к добавлению синтаксической конструкции «super()». Таким образом, для класса генерируется основной блок, имя класса и конструктор со свойствами класса. Создается файл *.js, в котором содержится код сгенерированных классов.

Фрагмент JS-кода представлен на рис. 2.

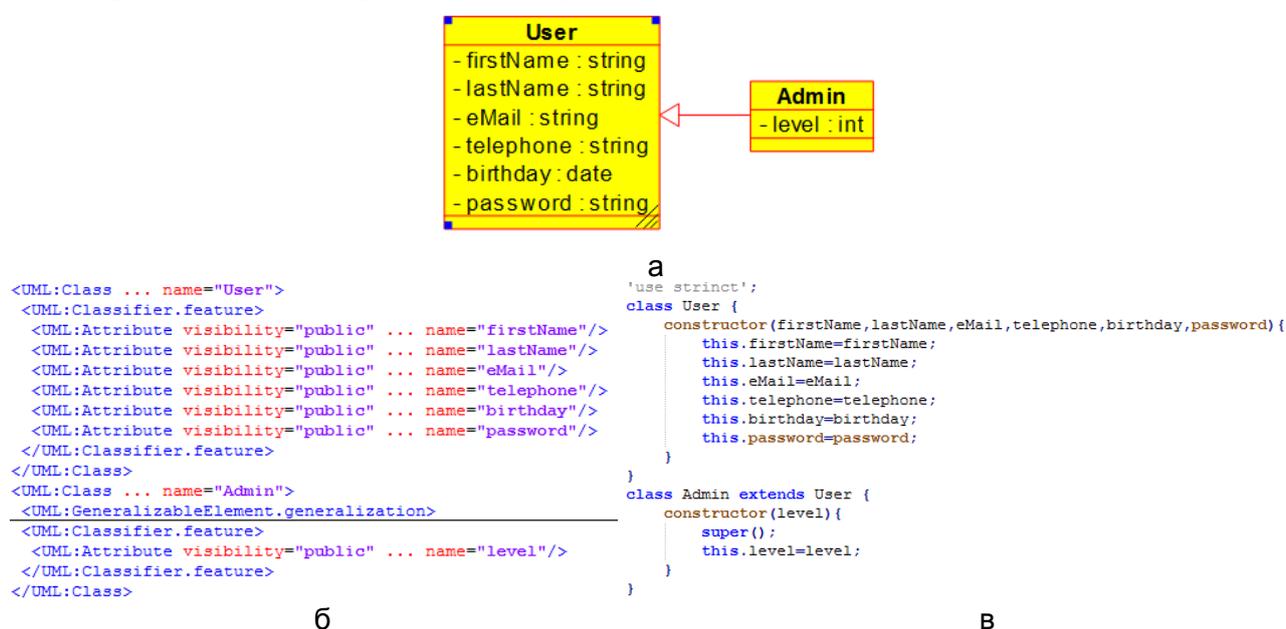


Рис. 2. Фрагменты диаграммы классов (а); XMI-документа (б) и сгенерированный JS-код (в)

В работе был рассмотрен метод генерации JS-кода на основании полученного от редактора UML представления моделей в формате XMI.

JavaScript обладает динамической типизацией, поэтому необходимо программно задать возможность контролировать типы, заданные в UML-модели.

К развитию данного метода генерации JS-кода необходимо добавить возможность генерировать код на основании диаграммы активностей, это позволит расширить возможности представленного генератора и создавать код методов класса.

Литература

1. Бузовский О.В., Алещенко А.В. Разработка системы генерации кодов по графическим схемам алгоритма с промежуточным языком трансляции // Информационные технологии – 2015. – № 4/2(24). – С. 15–19.
2. Стандарт ECMA-262 [Электронный ресурс]. – Режим доступа: <http://www.ecma-international.org/publications/standards/Ecma-262.htm>, своб.
3. Глухов Р.И., Елисеев А.С. Разработка web-редактора UML-моделей // Инновационные технологии: теория, инструменты, практика. – 2015. – Т. 1. – С. 112–118.

4. Спецификация ХМІ [Электронный ресурс]. – Режим доступа:
<http://www.omg.org/spec/XMI/>, своб.



Жукова Александра Валерьевна

Год рождения: 1983

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра управления и права, группа № S4115

Направление подготовки: 38.04.02 – Менеджмент

e-mail: alex3ukova@gmail.com

УДК 005.95/.96

**СОЦИАЛЬНО-ПСИХОЛОГИЧЕСКИЕ И МОРАЛЬНЫЕ МЕТОДЫ ПОВЫШЕНИЯ
ЭФФЕКТИВНОСТИ СИСТЕМЫ СТИМУЛИРОВАНИЯ ТРУДОВОЙ
ДЕЯТЕЛЬНОСТИ ПЕРСОНАЛА В УСЛОВИЯХ ЭКОНОМИЧЕСКОЙ
НЕСТАБИЛЬНОСТИ**

Жукова А.В.

Научный руководитель – к.психол.н., доцент Карпова Е.А.

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе проведено сравнительное исследование нематериальных способов мотивации персонала, выделены условия и типы работников, для которых эти меры подходят.

Ключевые слова: виды мотивации, развитие персонала, компетентность, исследование мотивации, нематериальная мотивация.

Даже если грамотному начальнику удалось сформировать коллектив из настоящих профессионалов, в конечном итоге он проиграет, если будет забывать о том, что персонал нужно постоянно мотивировать. Мотивация на выполнение работы ликвидирует «наплевательское» отношение к своим обязанностям у персонала.

Мотивация психологически делится на два типа: во-первых, стимул (как позитивный тип мотивации); во-вторых, угроза санкций (как негативный тип мотивации). Образно два этих типа мотивации определяются как «кнут и пряник» [1]. Мотивирование персонала бывает также материальным и нематериальным. Нематериальная мотивация – это система стимулов исключительно неденежного характера.

В работе речь пойдет о позитивном типе нематериальной мотивации. У руководителей и владельцев бизнеса часто возникает желание внедрить систему нематериальной мотивации, чтобы иметь возможность стимулировать работу сотрудников, не тратя на это деньги [2]. В нынешних экономических условиях это становится особенно актуальным.

Многие руководители, заставшие советскую эпоху, используют старый добрый метод «доски почета». Но уже мало кого из работников это вдохновляет на подвиги. Или, например, встречается такое мотивирование: лучший продавец месяца будет ужинать за одним столом с советом директоров. Материальной ценности в этом вовсе нет. Его дети не будут от этого лучше одеты, семья не сможет купить от этого машину. Но шаблонность мышления часто не позволяет проанализировать этот факт с точки зрения полезности для организма [3]. В этой связи многие ответственные и амбициозные сотрудники начинают стараться изо всех сил для достижения призрачного результата.

Рассмотрим более полезные для работника способы мотивирования, где за рабочие успехи ему дают возможность обучаться и развиваться.

Средства моральной и социально-психологической стимуляции должны приносить персоналу пользу. К таким способам относят организационные и морально-психологические средства мотивации. Примером таких средств может служить участие сотрудников компании в конкурсах профессионального мастерства.

Конкурсы профессионального мастерства способны актуализировать саморазвитие личности, замотивировать человека на желание стать компетентным сотрудником либо доказать свой профессионализм. Особенно ценно проведение конкурсов профессионального мастерства среди представителей тех профессий, которые предполагают работу руками – результат в таком случае виден предметно, работу можно в буквальном смысле потрогать и увидеть.

Очень помогают вовлечению игровые элементы в процессе работы, в совещаниях, в подведении итогов – это прекрасный способ развеять скуку и поднять уровень лояльности к компании [4].

Необходимо уделить внимание социальной активности, запустить в компании проекты по развитию, привлечь в них сотрудников. Смена деятельности, активное взаимодействие с другими сотрудниками в непривычном функционале помогут помочь мотивировать сотрудников и сделать их вовлеченными [4].

Людей важно благодарить и дарить им радость. Атмосфера в стенах офиса имеет огромное значение для психического здоровья коллектива [3]. Хорошо, когда меньше криков и неконтролируемых эмоций, а больше улыбок. Благодарить персонал логично, он зарабатывает для фирмы деньги.

Полезны разговоры с коллегами, советы, обсуждение планов фирмы. Руководителю и менеджеру желательно быть открытыми и позитивно настроенными, общаясь с работниками. Тогда легче выявлять демотивированных, выгоревших сотрудников и вовремя их «лечить».

К нематериальным стимулам можно еще отнести свободный или гибкий график работы, дополнительные выходные, благоприятную атмосферу в рабочем коллективе, похвалу руководителя, возможность продвижения по службе, отсутствие жесткого дресс-кода, возможность открыто говорить с руководителем о своих трудностях.

Проведите оценку компетенций сотрудника и определите, на своем ли месте он находится [3]. Его эффективность возрастет в геометрической прогрессии, как только он начнет реализовывать свой потенциал. Например, перегруженный работой копирайтер может угасать как специалист только потому, что никто не попытался разглядеть в нем прекраснейшего аналитика [3]. Важно провести анкетирование работников при приеме на работу, либо потом, если это не было сделано вовремя. Так можно выяснить сильные стороны работника и использовать их на благо фирмы, не создавая у него перегруженности той работой, к которой он менее способен. Работу, которая соответствует склонностям и способностям, люди чаще всего делают с энтузиазмом. Напротив, то, к чему человек не склонен, порой вызывает сильное нервное напряжение и, как следствие, выгорание.

Менеджеры считают, что способны понимать, предсказывать и контролировать поведение сотрудников [5]. Но они должны знать, что сотрудники реально хотят от своих рабочих мест. Крайне важно для менеджера понять, каковы настоящие мотивы служащих, а не основывать свою работу лишь на предположениях и догадках. Работников желательно опросить о том, что они думают о конкретных ситуациях, не давая свою точную оценку их потребностей, поскольку толкования и интерпретация менеджера по кадрам могут исказить то, что есть в действительности. Таким образом, повысить эффективность труда можно, получая лучшее представление о реальных потребностях сотрудников [1].

Позитивная нематериальная мотивация применима не ко всем работникам и не во всех условиях. Базовые потребности персонала должны быть удовлетворены. Не имеет смысла при минимальной зарплате мотивировать работников на самоотверженный труд. Это уже будет называться самопожертвованием, а на него идут только высоконравственные люди и ради высоких целей: защита семьи, Родины, помощь всему человечеству. Люди не займутся работой в ущерб своему здоровью ради чьей-то чужой, пусть даже и большой, фирмы. Для выполнения работ с очень примитивными трудовыми операциями, не требующими интеллектуальных усилий применимо мотивирование в рамках теории Х. Д. МакГрегора.

Согласно этой теории считается, что большинство людей ленивы, безответственны и хотят быть управляемыми, они не хотят принимать ответственность на себя и больше всего ценят безопасность и стабильность [1]. Теория Х провозглашает, что люди мотивированы финансовыми средствами и угрозой наказания. Менеджеры этого направления директивны, они строго контролируют подчиненных.

Д. МакГрегор подчеркивал, что директивные методы не применимы к людям, чьи физиологические потребности и потребности в безопасности удовлетворены, в то время как потребности в уважении и в самоактуализации становятся для них все более важными. Директивные методы не применимы к интеллектуальным работникам [6].

В современной литературе по управлению утверждается, что работники интеллектуального труда представляют собой особый вид рабочей силы, для эффективного управления которым необходимо переосмыслить традиционные подходы к управлению. Их структура мотивов трудовой деятельности другая, чем у неквалифицированных рабочих [7]. Интеллектуальные работники ищут любой повод, чтобы удовлетворить свои социальные потребности и потребности в самоактуализации, если у них достаточно денег для удовлетворения базовых потребностей.

В тех организациях высок уровень производительности, где люди приходят на работу с удовольствием, где работа удовлетворяет их высшие потребности [2]. Теория Y Д. МакГрегора считает, что люди не ленивы, и что на них можно рассчитывать. Люди могут быть самостоятельными и очень творческими, если они мотивированы должным образом. Согласно теории мотивации Д. МакКлелланда, людей привлекает возможность быть принадлежащими к престижному сообществу, успешной корпорации, признание их достижений и обладание властью [7].

Высшее управленческое искусство заключается не в том, чтобы заставить человека действовать нужным образом, а в создании таких условий, когда он сам придет к выводу о целесообразности подобного действия [5].

Оценить эффективность процесса мотивации труда можно с помощью количественных и качественных показателей [5]. Желательно проводить мониторинг регулярно, чтобы иметь обратную связь.

Литература

1. Кибанов А.Я., Баткаева И.А., Ивановская Л.В. Управление персоналом организации. – 4-е изд., доп. и перераб. – М.: НИЦ ИНФРА-М, 2014. – 695 с.
2. Emma Seppala. To Motivate Employees, Do 3 Things Well [Электронный ресурс]. – Режим доступа: <https://hbr.org/2016/01/to-motivate-employees-do-3-things-well>, своб.
3. Шишкина И. Кейс: как зажечь эмоционально выгоревших сотрудников [Электронный ресурс]. – Режим доступа: <https://ubo.ru/articles/?cat=107&pub=3696>, своб.
4. Андреева Т.Е. Особенности мотивации работников интеллектуального труда: первичные результаты исследования // Российский журнал менеджмента. – 2010. – Т. 8. – № 2. – С. 47–68.

5. Якунин В.И., Сулакшин С.С., Багдасарян В.Э. Восстановление инструментария мотиваций в государственном управлении России. – М.: Научный эксперт, 2009. – 224 с.
6. McGregor D & Cutcher-Gershenfeld J. The human side of enterprise. – New York: McGraw Hill, 2006. – 256 p.
7. Employee motivation and organizational performance [Электронный ресурс]. – Режим доступа:
http://ftp.repec.org/opt/ReDIF/RePEc/rse/wpaper/R5_5_DobreOvidiuIliuta_p53_60.pdf, своб.

**Журавлев Виктор Олегович**

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4207Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: vzhuravlev15@gmail.com

УДК 004.652

**ТЕОРЕТИЧЕСКИЕ ОСНОВЫ NODE.JS ФРЕЙМВОРКОВ: ПАТТЕРНЫ
АСИНХРОННОГО КОДА «CALLBACK», «COROUTINE», «PROMISE»,
«ASYNC/AWAIT»****Журавлев В.О.****Научный руководитель – к.педагог.н., доцент Государев И.Б.**

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе представлен краткий обзор фреймворков Express и Koa. Описаны приемы обращения с асинхронным кодом с использованием новых возможностей спецификации ECMAScript 6.0.

Ключевые слова: JavaScript, ECMAScript 2015, ECMAScript 6, callback, promise, generator, coroutine, async/await.

С момента появления Node.js вызывал растущий интерес в среде разработчиков. Сильные стороны Node.js – это возможность обращения к серверной части и огромным объемам данных удобным и эффективным способом, а также одновременная обработка огромного количества соединений с большой пропускной способностью. Это инициировало многочисленные разработки различных фреймворков для создания более эффективных инструментов на основе данной технологии [1–4].

На сегодняшний день сформировался широкий выбор фреймворков, самые примечательные из которых – это Express и Koa, созданные организацией Node.js Foundation.

Express.js в первую очередь использует первичные принципы и подходы Node.js, расширяя функциональные возможности в значительной степени и создавая новые удобные функции. Данный фреймворк является минималистичным и очень надежным инструментом для создания одностраничных и многостраничных сайтов, а также интерфейсов прикладного уровня (API), в то же время он гарантирует максимальную возможную производительность.

Koa.js предоставляет мощный каркас сервера для создания эффективных веб-приложений и интерфейсов прикладного уровня (API). Фреймворк был написан полностью с нуля, во многом он похож на Express, но есть одно существенное отличие –

больше не используются функции обратного вызова (callback). Он не обременен поддержкой legacy-кода, поэтому код был переписан на современный ES6 (ES2015). Вместо функций обратного вызова в первой версии используются так называемые корутины – это функции библиотеки Co, использующие генераторы и обещания (Promise). Но разработчики не стали останавливаться на этом и выпустили вторую версию фреймворка, написанного на стандарте ES2016 и ES2017.

Создание фреймворка Koa.js на экспериментальной на данный момент спецификации было необходимо для предоставления возможности написания более понятного, производительного и минималистичного кода разработчиками. Это было достигнуто благодаря избавлению от бесконечно вложенных функций обратного вызова.

Одной из сильнейших сторон JavaScript является работа с асинхронным кодом. Для решения асинхронных задач используется обработка событий, что является универсальным подходом или с помощью менее удобного способа – функций обратного вызова (callback).

Самый существенный недостаток функций обратного вызова с точки зрения организации кода, это образование: «callback hell» или «pyramid of doom». Проблема состоит в том, что в параметрах вызываемой функции есть параметр, который является также функцией обратного вызова – и все это может продолжаться до бесконечности (рис. 1).

```
asncFunction1(function(err, result) {
  asncFunction2(function(err, result) {
    asncFunction3(function(err, result) {
      asncFunction4(function(err, result) {
        asncFunction5(function(err, result) {
          // do something useful
        })
      })
    })
  })
})
```

Рис. 1. Многочисленные уровни вложенности функций обратного вызова

Из-за этого образуются многочисленные уровни вложенностей блоков, что снижает интуитивное понимание логики работы при больших объемах кода и увеличивает степень запутанности функций обратного вызова, а это влечет за собой снижение скорости разработки и повышает вероятность возникновения ошибок.

Новая спецификация ECMAScript 6.0 (ES2015), вышедшая в 2015 году, предоставляет много инструментов для решения проблемы многочисленных функций обратного вызова.

Первым инструментом, представленным в новой спецификации для работы с асинхронностью, стали обещания (Promise). Данное нововведение помогает изменить подход к функциям обратного вызова и позволяет удобно выстраивать в цепочку последовательные асинхронные вызовы. Работа с Promise сводится к взаимодействию с состояниями. Вначале обещание находится в состоянии ожидания (pending), после выполнения существует еще два состояния: завершен успешно (fulfilled) и завершен с ошибкой (rejected) (рис. 2).

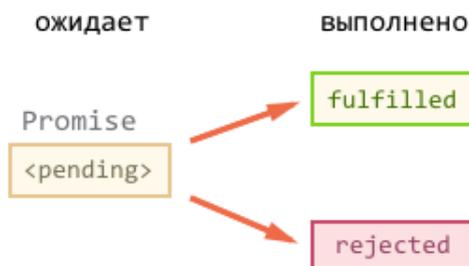


Рис. 2. Возможные состояния ожидания, после выполнения инструкции

Оба состояния возвращают обещания, что позволяет выстраивать цепочки запросов с обработкой данных.

Следующим инструментом являются функции – генераторы (Generator). Данный инструмент предоставляет очень мощный механизм написания чистого и понятного асинхронного кода. В основе работы генератора лежит использование стекового кадра, позволяющего выполнять передачи аргументов и выделение временной памяти с использованием системного стека. Во время выполнения генератора нет никакой параллельности, действия происходят синхронно и однопоточно. Используя генераторы, необходимо писать функции для последовательного перебора `next()` методов или циклы. Есть уже готовые библиотеки с готовыми реализациями переборов, а также обладающих большим количеством возможностей.

Одна из самых известных библиотек – это `Co`. Библиотека объединяет различные функции-обертки (корутины) над обещаниями и генераторами, позволяющие писать простой и понятный псевдосинхронный код.

Корутина (coroutine) – это функция, построенная на основе генераторов, позволяющая контролировать и многократно передавать как вовне так и внутрь функции данные. Каждая созданная корутина является специальной «функцией-чернорабочим» которая запускает генератор и последовательными вызовами `next()` получает из него промисы.

Новый инструмент `async/await` работающий на обещаниях и генераторах, избавляет от необходимости использования `then()` методов обещаний. Алгоритм работы построен на переборе последовательности генератора (из `await` выражений), упаковывающего каждый элемент последовательности в свое обещание, после чего формируются цепочки. Благодаря такому удобному «синтаксическому сахару» над генераторами и обещаниями, писать асинхронные функции через `async/await` гораздо удобнее: получается меньше кода и пропадает необходимость в оборачивании промисов в генераторы.

После принятия ECMAScript 6.0 было исправлено много ошибок, добавлены классы, стрелочные функции, увеличена производительность, но самое главное нововведение – это обещания и генераторы – инструменты для работы с асинхронными функциями и запросами. На рис. 3 представлено сравнение затраченного времени на работу новых инструментов и функций обратного вызова.

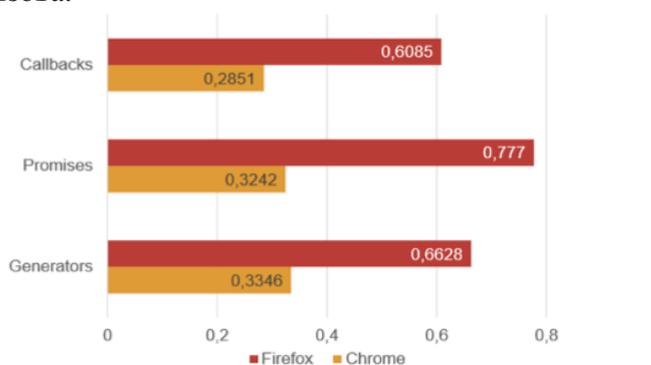


Рис. 3. Сравнение затраченного времени на работу асинхронного кода с помощью новых инструментов и функций обратного вызова

Изучив данные, можно сделать выводы, что новые функциональные возможности немного более медленные. Но эти минимальные отставания во времени незначительны по сравнению с предоставляемыми возможностями, а также их решением проблемы вложенностей функций обратного вызова.

Функция Async/Await, благодаря удобной реализации, способна заменить генераторы и обещания. Но она все еще не включена в официальную спецификацию ECMAScript.

Решение о выборе конкретного инструмента должно быть принято исходя из рациональности и требований в каждом отдельно взятом новом проекте.

Литература

1. ECMAScript 2017 Language Specification [Электронный ресурс]. – Режим доступа: <https://tc39.github.io/ecma262/>, своб.
2. Браун И. Веб-разработка с применением Node и Express. Полноценное использование стека JavaScript. – Питер, 2017. – 336 с.
3. Async Functions for ECMAScript [Электронный ресурс]. – Режим доступа: <https://github.com/tc39/ecmascript-asyncawait>, своб.
4. Современные возможности ES-2015 [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/es-modern>, своб.



Захаревич Елизавета Андреевна

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4205

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: zaharevich.elizaveta@mail.ru

УДК 004.946

ВИРТУАЛЬНЫЙ ТУР КАК НАГЛЯДНЫЙ МЕТОД ПРЕЗЕНТАЦИИ ПРОЕКТА

Захаревич Е.А.

Научный руководитель – д.т.н., профессор Погорелов В.И.

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

Цель работы – рассказать о виртуальном туре, как наиболее наглядном методе презентации проектов, его преимуществах, рассмотреть сферы и отрасли, в которых можно использовать технологию 3D-туров и панорам. Также в работе рассмотрена технология создания виртуальных туров и наиболее эффективные и популярные программные приложения для быстрого создания панорам и 3D-туров.

Ключевые слова: виртуальный тур, 3D-тур, сферическая панорама.

Сферические 3D-панорамы и виртуальные туры – это современный и эффективный метод презентации пространства. Основной средой размещения виртуальных туров является интернет-пространство, однако 3D-панорамы распространены и на оффлайн-носителях (дисках, флэш-носителях, обучающих материалах и других видах презентаций проектов). 3D-панорама – это фотография, охватывающая все пространство

вокруг одной точки на 360°. Панорама собирается из нескольких фотографий в одно бесшовное изображение, чем и достигается эффект присутствия в центре изображения.

С помощью кнопок навигации можно просматривать пространство вокруг, приближать или удалять предметы и объекты, изучать общий план. Несколько сферических 3D-панорам могут быть соединены друг с другом в виртуальный тур (или 3D-тур). В таком туре методом нажатия на кнопки навигации можно перемещаться из одной 3D-панорамы в другую. Для лучшей ориентации посетителей виртуальный тур может сопровождаться картой или радаром. 3D-тур идеально подходит для презентации зданий и интерьеров, экскурсий по городам и музеям. Таким образом, находясь за компьютером, можно перемещаться из одной комнаты в другую, рассматривать объекты недвижимости, а также посещать различные достопримечательности, музеи, любые уголки Земли и даже спускаться на морские глубины! (Например, в марте 2012 г. на подводном спускаемом аппарате российского производства «Мир-2» в составе научной группы фотографам удалось опуститься на дно самой глубокой в мире Марианской впадины глубиной около 11 км, которая могла бы с легкостью «проглотить» гору Эверест).

Несомненным плюсом виртуальных туров является то, что для их показа не нужно дополнительно устанавливать никаких программ, достаточно иметь Интернет-браузер, поддерживающий HTML5 или Flash. Сферические панорамы и виртуальные туры, созданные с помощью современного оборудования, прекрасно отображаются не только на компьютерах, но и на мобильных устройствах, что, несомненно, делает их более доступными и удобными еще большему количеству пользователей.

Виртуальный тур дает случайному посетителю сайта точное представление об объекте, будь то дизайн-проект интерьера, объект недвижимости или целая экскурсия по достопримечательностям, ведь даже самые подробные описания и фотографии не в силах передать полную картину. С помощью виртуального тура создается «эффект присутствия»: клиент может рассмотреть объект в мельчайших деталях, можно предмет отдалить или приблизить, осмотреть под разными углами. Также нет никаких сомнений, что такое путешествие запомнится намного лучше, чем серия даже самых качественных фотографий. Следует отметить, что виртуальные туры и виртуальные панорамы – это эффективный инструмент маркетинга в социальных сетях. Такие панорамы и виртуальные туры могут быть интегрированы в Facebook, ВКонтакте, Google+ на страницу или в группу компании, пользователи не только смогут просматривать их, но и добавлять к себе на страницу.

Можно без преувеличения сказать, что виртуальный тур – это находка для настоящих профессионалов в области дизайна интерьеров и архитекторов. Каждый дизайнер знает, что потенциальному клиенту очень важно сначала увидеть предыдущие работы. Однако плоские фотографии не несут всей полноты проделанной работы. Более того, по обычным фотографиям невозможно судить об атмосфере, созданной дизайнером в помещении. Благодаря виртуальному туру дизайнер имеет возможность продемонстрировать свои лучшие проекты, дополняя их комментариями и пояснениями. 3D-тур имеет возможность продемонстрировать потенциальному клиенту каждую деталь интерьера, выгодно подчеркнуть его и максимально точно передать общую идею созданного дизайна.

Также с большим успехом можно использовать 3D-тур в экскурсионном и туристическом бизнесе, ведь большинство пользователей выбирают места отдыха или посещения именно по визуальной информации, которую они получают в Интернете. Причем многие назвали 3D-панорамы и виртуальные туры очень полезными при выборе места посещения или отдыха. Виртуальные экскурсии, просмотры интерьеров замков, исторических зданий – хорошая реклама, привлекающая посетителей. Кроме этого 3D-

туры дают возможность просматривать экспонаты, которые не могут быть доступны реальным посетителям.

Отели, санатории, спа-салоны, рестораны могут использовать 3D-панорамы и виртуальные туры, которые могут предоставить клиентам возможность увидеть убранство и уют в помещениях, а также красоту окружающей территории.

Производители транспортных средств тоже могут использовать 3D-туры. С помощью сферической 3D-панорамы покупатель может просмотреть интерьер выбранной машины еще перед посещением салона. Виртуальная панорама позволяет идеально демонстрировать даже интерьеры с очень ограниченным размером.

3D-панорамами и виртуальными турами могут быть оснащены информационные центры и сайты городов. Виртуальные туры помогут туристам и гостям сориентироваться в городе, покажут наиболее интересные маршруты и достопримечательности.

Изготовление качественной виртуальной панорамы – процесс, состоящий из нескольких этапов: фотосъемка объекта, обработка полученных изображений и конечная сборка виртуального тура.

Итак, сначала объект фотографируется. Для съемки используется цифровая зеркальная камера со сверхширокоугольным объективом или объективом «рыбий глаз». Камерой, прикрепленной на стабильном штативе, снимается серия кадров. Для создания сферической панорамы необходимо минимум два кадра, но такая панорама будет не очень хорошего качества. Обычно для создания панорамы лучшего качества снимают от 4 кадров.

Следующим шагом по созданию **3D-панорам** является обработка исходных снимков и их соединение в панораму. Существует множество программ, предназначенных для склейки фотографий: AutoPano Pro, PTGui Pro, Hugin, PhotoFit, Pixtra OmniStitcher, Panorama Factory, Pana Vue ImageAssembler, Smartblend, Panorama Tools. Даже компания Autodesk, создатель 3ds Max, отметилась в этом сегменте созданием Autodesk Stitcher Unlimited. Однако в большинстве вышеперечисленных программ удобно работать, если есть профессиональная аппаратура и широкоугольный светосильный объектив. В другом случае без алгоритмов автоматизации не обойтись. Несомненными лидерами являются Kolor Autorano Giga (кстати, в марте 2010 г. с помощью программы Autorano Giga была создана крупнейшая на тот момент цифровая фотография – вид Парижа с разрешением в 26 гигапикселей) [1], Hugin, PTGui Pro [2].

Следующий шаг – сборка виртуального тура с помощью определенных программ. Существует много программ для их создания: Kolor Panotour Pro, Tourweaver, Pano2QTVR free, Pano2VR, Photo Warp, Panoweaver, freeDEXpano, JATC, Panorama2Flash, Pixtra TourMaster, Flash Panorama Player. Однако в этой сфере бесплатного софта очень мало, и наиболее популярными программами, у которых соотношение цена-качество наиболее адекватно являются следующие: Kolor Panotour Pro, лидер всей индустрии по созданию виртуальных туров, и Easypano Tourweaver. Эти программы отличаются удобным графическим интерфейсом и интуитивным управлением проектом. Результатом сборки является набор файлов, которые не составят большого труда интегрировать в страницу сайта. Однако почему-то разработчики Easypano проигнорировали такую платформу как Macintosh. На Macintosh туры не отображаются, хотя на устройствах с iOS на борту все работает [3]. При создании виртуального тура в каждую сферическую панораму добавляются элементы навигации. Навигация может включать в себя схему территории с активными точками переходов; радар, указывающий направление взгляда; всплывающее превью следующей панорамы в виде мини-фотографии или мини-панорамы; выпадающее меню с

названиями всех панорам в виртуальном туре; всплывающие информационные блоки; различные интерактивные элементы; дикторский голос и музыку. В конце навигации готовая 3D-панорама или виртуальный тур загружаются на сайт Заказчика или записываются на носители.

На протяжении долгого времени наиболее распространенным средством просмотра **3D-панорам и виртуальных туров** являлся Flash Player компании Adobe. Популярные ранее QuickTime, Deval VR, Java и другие форматы используются гораздо реже. Однако теперь более популярным становится формат HTML5, так как для просмотра панорам, построенных на технологии HTML5, не требуется установка плагина. Воспроизведение сферических панорам и виртуальных туров осуществляется полностью силами браузера.

Таким образом, учитывая все вышесказанное, можно сделать вывод о том, что 3D-панорамы объектов, виртуальные туры пришли на смену обычным фотографиям, и технологию виртуальных туров можно применять во многих сферах и отраслях.

Литература

1. Paris 26 gigapixels [Электронный ресурс]. – Режим доступа: <http://www.paris-26-gigapixels.com/index-en.html>, своб.
2. Master360: сервис для виртуальных туров [Электронный ресурс]. – Режим доступа: <http://master360.ru/blog/urok-3-vybor-programmnogo-obespechenija>, своб.
3. Mediakritika [Электронный ресурс]. – Режим доступа: <http://mediakritika.by/article/2921/sobiraem-panoramy>, своб.



Захарова Анастасия Олеговна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4106
Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: aozakharova18@gmail.com

УДК 004.92

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ГЕЙМ-ДИЗАЙНА НА РАЗЛИЧНЫХ ПЛАТФОРМАХ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

Захарова А.О.

Научный руководитель – д.т.н., профессор Погорелов В.И.

Работа направлена на изучение истоков и поиск путей дальнейшего развития гейм-дизайна как вида проектной деятельности. В работе представлены результаты сравнительного анализа различных игровых платформ, а также выделены основные этапы проектирования дизайна VR-игр. Включены дополнительные источники, необходимые для более полного изучения темы, послужившие информационным базисом при проведении исследования.

Ключевые слова: гейм-дизайн, виртуальная реальность, VR-платформы, VR-игры, шлем виртуальной реальности.

Виртуальная реальность погружает человека в трехмерное окружение при помощи специальных приспособлений: шлемов, очков или VR-платформ. По этой причине довольно часто используется термин iVR (immersive VR – виртуальная реальность с погружением). Для того чтобы обособиться от виртуальных миров, которые в начале 90-

х годов прошлого века своей технологией привлекли к себе всеобщее внимание, но не смогли предоставить ожидаемых ощущений, в связи с нехваткой технических мощностей и отсутствием глубоких знаний в этой области [1–5].

Однако тогда было положено начало тому, что проекты виртуальной реальности (VR) смогут не только создавать концептуально новые рынки, но и расширять уже имеющиеся. На сегодняшний день существует 9 потенциальных сфер применения технологий VR:

1. видеоигры;
2. мероприятия в прямом эфире;
3. кино и сериалы;
4. продажи;
5. образование;
6. здравоохранение;
7. военная промышленность;
8. продажи недвижимости;
9. проектирование.

Очевидно, сфера развлечений остается лидером по использованию технологии VR, поскольку игровая индустрия стремительно движется к оценке в 30 млрд долл. США и, по оценкам, экспертов вырастет еще в 60 раз уже к 2018 году.

Итак, какова же задача виртуальной реальности? Задача – заставить пользователя почувствовать себя в другом мире, дать возможность уйти от реальности. В связи с этим при создании VR-игр огромное внимание уделяется эффекту присутствия. Этот аспект отличает дизайн VR от дизайна обычных видеоигр. Но помимо эффекта присутствия, внимание также уделяется нелинейности повествования, не вызывающему тошноты перемещению и графической оптимизации.

На основе изученной литературы, а также электронных ресурсов было выделено 7 основных аспектов проектирования дизайна VR-игр:

1. роль среды;
2. атмосфера;
3. особенности местности;
4. представление среды пользователю через гамму звуков;
5. обучение пользователя взаимодействию с объектами;
6. контекстный прицел;
7. интерактивные объекты.

На каждом из вышеперечисленных аспектов необходимо сосредоточится при проектировании дизайна VR-игр, поскольку именно от этих составляющих зависит эффект присутствия. Пройдя все основные этапы проектирования VR-игры, возникает необходимость представить ее пользователю через специальные технические средства. Такими средствами, как уже говорилось в начале, выступают шлемы и платформы. Благодаря им пользователь может погрузиться в спроектированный разработчиками мир. Конечно, поставленной задачи погружения пользователя можно добиться даже малыми средствами, однако, для достижения максимального эффекта присутствия, необходимы более совершенные приспособления.

Такими приспособлениями в наше время являются VR-платформы, периферийное оборудование в виде всенаправленной беговой дорожки для взаимодействия с виртуальной реальностью в играх. Ниже приведена сводная сравнительная таблица самых популярных VR-платформ, их преимущества, недостатки и цена.

Таблица. Сравнение самых популярных VR-платформ

Модель/ характери- стика	Описание	Преимущества	Недостатки	Цена (руб.)
PRODE	Является платформой для передачи действий от пользователя к персонажу, за которого играет пользователь	Каждый шаг, прыжок или приседание будет распознан компьютером как команда к действиям, по тому же принципу, как если бы эти команды были отданы с помощью клавиатуры	Требуются отдельные контроллеры и обувь	80000
ROVR	VR-платформа для профессионалов, разработчиков контента и обычных пользователей	1. Маленький вес относительно других моделей. 2. Позволяет поворачиваться на 360 градусов, при этом непринужденно шагая как в реальном, так и в виртуальном мире.	Требуются отдельные контроллеры и специальная обувь	68000
VIRTUIX OMNI	Революционная беговая дорожка Omni от компании Virtuix	Дорожка выступает в роли большого джойстика. Все действия с точностью фиксируются и передаются персонажу	Необходим VR-шлем	100000
CYBERIT И VIRTUALI ZER	Платформа, обладающая расширенным функционалом, который разработан специально для коммерческого использования. Среди главных отличий от домашних версий можно отметить инновационный механизм крепления, активируемый в один шаг. Он обеспечит безопасность при использовании даже новичкам	1. Регулируемые фиксаторы позволяют настроить беговую дорожку под любой рост и позволят прыгать и приседать геймеру во время игры. 2. Датчик скорости бега определяет, каким шагом идет пользователь. 3. Не нужна специальная обувь.	Не выявлено	140000

Модель/ характери- стика	Описание	Преимущества	Недостатки	Цена (руб.)
OCULUS RIFT	Очки виртуальной реальности	1. Огромная область видимости. 2. Безопасность для глаз.	1. Единственн ый датчик для отслежива- ния движения – что замедляет синхрониза- цию и работу девайса. 2. Расширение 1080×960 не дает максимально объемного изображения . 3. Требуется обязательно- го проводного подключени я к игровому ПК или консоли.	40000

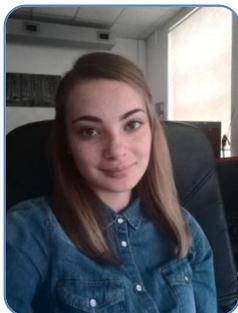
Таким образом, основной трудностью, возникающей при разработке VR-игр, является стоимость платформ виртуальной реальности. Не каждый пользователь может позволить себе приобрести VR-платформу для личного пользования, что, в свою очередь, приводит к тому, что крупнейшие игровые разработчики не рискуют вкладываться в разработку новой серии игр.

Однако данная тема довольно актуальна, уже сейчас прослеживается позитивная тенденция в развитии проектирования VR-игр и популяризации среди потребителей игровой индустрии. В дальнейшем проблема себестоимости VR-платформ будет решена, что даст новый толчок в развитии дизайна VR-игр.

Литература

1. Основы создания виртуальной реальности [Электронный ресурс]. – Режим доступа: <https://www.progamer.ru/dev/creating-vr.htm>, своб.
2. 9 сфер применения виртуальной реальности: размеры рынка и перспективы [Электронный ресурс]. – Режим доступа: <https://vc.ru/p/vr-use>, своб.
3. Creating Virtual Reality Games: The Fundamentals [Электронный ресурс]. – Режим доступа: http://www.gamasutra.com/view/feature/192810/creating_virtual_reality_games_.php?page=4, своб.
4. Dille F., Platten J. The ultimate guide to video game writing and design. – New York: Random House, Inc., 2007. – 272 p.

5. Mitchell B. Game Design Essentials. – Indianapolis, Ind.: John Wiley & Sons, 2012. – 320 p.



Зиборова Ирина Андреевна

Год рождения: 1992

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4206

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: irisha.ziborova@gmail.com

УДК 004.514

**МЕТОДЫ И ЭТАПЫ СОЗДАНИЯ ДИЗАЙНА И ПРОТОТИПА
ПОЛЬЗОВАТЕЛЕ-ОРИЕНТИРОВАННЫХ ИНТЕРФЕЙСОВ**

Зиборова И.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 рассмотрено несколько атрибутов Canvas в отношении их влияния на производительность рендеринга. Описано экспериментальное приложение, которое используется для проведения тестов производительности. Приведены результаты тестов рендеринга, а также сравниваются ожидания пользователей с полученной производительностью. Затронуты возможности использования Canvas с Web Workers.

Ключевые слова: HTML5, холст, анимация, графические эффекты, время рендеринга, тест производительности, ожидания пользователей.

В процессе создания любого интерфейса большую роль играют используемые технологии и то, как они позволяют сформировать интерфейс, отвечающий новым стандартам дизайна и функционала, а также повысить удовлетворенность пользователя от работы с интерфейсом. HTML5, опубликованная консорциумом World Wide Web (W3C) в качестве рекомендации в октябре 2014 года [1] представляет собой совокупность многих технологий, которые используются при создании любого интерфейса. Одной из родных технологий HTML5 является Canvas. Он используется для рисования геометрических фигур, контуров и цветовых градиентов, и имеет функциональные возможности для манипуляций с изображениями. Canvas может также использоваться для создания анимации. Но Canvas не единственное HTML5-приложение для рисования. Рядом с Canvas стоит Scalable Vector Graphics (SVG [2]) и X3D [3], которые используются для 3-мерного моделирования.

С целью исследования сильных сторон Canvas по сравнению с другими технологиями было создано экспериментальное приложение. Экспериментальное приложение помогает оценить, как время рендеринга меняется при влиянии на Canvas следующих факторов:

- большое количество объектов (кривые, круги, ...);
- графические эффекты (прозрачность, тени);
- анимация.

Разработанное приложение образует взаимосвязи между городами при помощи кривых Canvas на карте мира. Для достижения визуальной эстетики используются различные графические эффекты, такие как прозрачность и тени (рисунок).

Учитывая пять переменных, в частности, с числом кривых: количество кругов, прозрачность, тени и анимации, были проведены 15 тестов для оценки влияния вышеуказанных переменных на время рендеринга элементов Canvas. Испытания были проведены на Mac OS X 10.11.6 в браузере Safari версии 9.1.2. Результаты тестов показали, что время рендеринга на 100 кривых Canvas не меняется, когда используются любые другие свойства: прозрачность, тени, дополнительные окружности или анимация. Тем не менее, существует возрастающая разница во времени рендеринга при использовании большего количества элементов. Особенно сильно на время рендеринга влияет наличие атрибута тени, а вот использование атрибута анимации, наоборот, значительно снижает время рендеринга. Этот результат получается благодаря расщепленным кривым Безье, которые используются для создания линейной прогрессирующей анимации.



Рисунок. Экспериментальное приложение – взаимосвязи между городами на карте мира

Результаты испытаний показали, что Canvas хорошо работает в отношении времени рендеринга при использовании большого количества элементов. Это особенно важно, так как пользователи имеют ограниченное терпение, когда речь идет о времени отклика веб-страницы:

- «47 процентов потребителей ожидают загрузку веб-страницы не более чем через две секунды или [...]»;
- 40 процентов потребителей не будет ждать более трех секунд [4]».

Однако в то время как рисунок Canvas проходит рендеринг, веб-сайт не отвечает на запросы пользователя. Решением невосприимчивости HTML5 во время вычислений является приложение Web Workers. С помощью Web Workers рендеринг может работать в фоновом режиме, не мешая пользовательскому интерфейсу. Это было бы большим преимуществом в применении Canvas с относительно длительным временем рендеринга, однако, на текущий момент Web Workers может только выполнять вычисления и не может напрямую взаимодействовать с DOM. Так что использование Web Workers для приложения Canvas не представляется возможным [5].

Так как HTML постоянно развивается, родные веб-технологии рисования становятся более стандартизированными и адаптированными к интеграции. Приведенный выше пример с Canvas и Web Workers показал, что в настоящее время ведутся работы, чтобы предоставить разработчикам еще более связанное с веб-технологиями решение для предоставления конечным пользователям быстрые, гибкие и графически насыщенные приложения.

Литература

1. Консорциум World Wide Web. HTML5. Рекомендация W3C. – 2014.
2. Джонсон Д.У. и Jankun-Келли Т.Дж. Изучение масштабируемости графически визуализированной информации // Материалы графического интерфейса – Канадского общества обработки информации.
3. Mozilla Developer Network / Использование Web Workers [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/ru/docs/DOM/Using_web_workers, своб.
4. Akamai Technologies / 2 секунды на отклик как новый критерий доступности страницы для ресурсов в электронной коммерции. – 2009.
5. Web3D Consortium, Inc. Information Technology / Компьютерная графика и обработка изображений – Расширенный 3D (X3D) кодирование. – Часть 1 – ISO/IEC 19776-1:2005, ISO/IEC, 2005.



Зуев Евгений Андреевич

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: carnage02@mail.ru

УДК 004.58

АНАЛИЗ ОНЛАЙН-КОМПИЛЯТОРОВ И ИНТЕРПРЕТАТОРОВ

Зуев Е.А.

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

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе проанализированы наиболее популярные электронные ресурсы, включающие в себя онлайн-компилятор, выявлены основные недостатки систем. На основе результатов проведенного анализа сформулированы требования к системам, выполнение которых будет способствовать повышению эффективности онлайн-обучения программированию на любом этапе обучения.

Ключевые слова: онлайн-компилятор, онлайн-интерпретатор, онлайн-обучение, обучение программированию, система самообучения.

В настоящее время актуально использование возможностей глобальной сети для реализации целей обучения. Одним из инструментов, помогающих начинающим практикантам в области программирования, является набирающий популярность онлайн-компилятор. В работе рассмотрены наиболее популярные электронные ресурсы, предлагающие возможность онлайн-компилирования. Рассматривая списки электронных ресурсов, попавших в первую десятку по запросу «онлайн-компиляторы» в поисковой системе Яндекс, наиболее популярными были выявлены следующие онлайн-среды разработки [1–3]:

- Ideone [<http://ideone.com/>] – был представлен 3 раза;
- GCC GodBolt [<http://gcc.godbolt.org/>] – был представлен 2 раза;
- OnlineCompiler [<http://www.onlinecompiler.net/>] – был представлен 2 раза.

Рассмотрим последовательно каждую из трех онлайн-сред.

В качестве языка программирования был выбран C++ – компилируемый язык программирования общего назначения, поддерживающий объектно-ориентированное программирование, который поддерживается во всех представленных онлайн-компиляторах.

В качестве программы для тестирования была составлена небольшая программа с демонстрацией ввода и вывода, листинг которой представлен ниже:

```
1. #include <iostream>
2. using namespace std;
3. int main() {
4. char x[20];
5. cout<< "введите x\r";
6. cin >> x;
7. cout<< x;
8. return 0;}
```

Ideone. Онлайн-компилятор и отладочный инструмент, поддерживающий более 60 языков программирования и их версий.

После ввода представленного выше программного кода и ввода в поле stdin текста Hello_ИТМО, сервис провел успешную компиляцию и предоставил в поле stdout две строки с текстом «введите x» и «Hello_ИТМО». Также на странице был представлен инструмент Share or Embed source code, со строкой, содержащий скрипт для вставки готового кода на сайт.

После умышленно допущенной ошибки в 7 строке кода, заключающейся в замене переменной «x» на необъявленную переменную «y», в поле stdout присутствовало уведомление в виде сообщения от компилятора, содержащее в себе отчет об ошибке, что является преимуществом сервиса.

Основным преимуществом сервиса является возможность просмотра результата отработавшей скомпилированной программы. Доступно скачивание файла кода в расширении, соответствующем языку программирования. Сервис позволяет указать данные для ввода, добавить к коду описание, а также поделиться им с другими пользователями. Поддерживает подсветку синтаксиса. После регистрации добавляется возможность сохранить файл на сервере, после чего продолжить написание кода. Также доступна вставка сохраненного кода на сайт через элемент интерфейса Share or Embed source code. Есть примеры готового кода и листинг других пользователей, которые доступны к просмотру.

К недостаткам сервиса можно отнести невозможность скачать откомпилированную версию программы в формате exe. Также на сервисе присутствует реклама, занимающая около 20% экрана.

GCC GodBolt. Электронный ресурс обладает значительным отличием от большинства онлайн-сред и инструментов разработки, заключающееся в том, что он никаким образом не позволяет увидеть результат работы откомпилированной программы или скачать готовый файл. Идея ресурса заключается в том, что он позволяет ввести код на языке C++ в одном окне, и выбрав требуемый компилятор увидеть скомпилированный результат программы в другом. Таким образом, экспериментируя, сервис позволяет изучить в действии и оценить работу огромного количества компиляторов, их версий, их реакцию на изменение кода, реакцию кода на смену компилятора, интерпретировать программный код в язык Assembler, поддерживает вывод в синтаксисе Intel.

Также есть возможность сравнения нескольких результатов, скомпилированных в различных компиляторах одной и той же программы. Сервис поддерживает подсветку синтаксиса, нумерацию строк. Присутствует виджет Share, выдающий HTML-строку для вставки примера с кодом на сайт. После умышленного ввода ошибки, сервис сразу ее находит и выдает сообщение для каждого открытого компилятора в поле с введенным кодом. В дополнение к небольшим плюсам ресурса можно отнести мгновенное нахождение ошибки и мгновенную компиляцию после ее исправления, также полное отсутствие рекламы, возможность увеличения шрифта в каждом окне, что может помочь слабовидящим, поддержка пяти примеров готовых программ.

Главными недостатками являются отсутствие возможности сохранить файл в формате `exe` и увидеть результат рабочей программы после ее сборки.

Online Compiler. Единственный рассмотренный сервис, поддерживающий возможность сохранить скомпилированную версию программы в активном формате, что является главным его преимуществом. Есть два варианта расширений активных форматов, для систем DOS/Windows и для Unix-подобных операционных систем.

После выбора C/C++, как одного из 5-ти поддерживаемых языков программирования, открывается онлайн-компилятор с заполненным полем введения кодом программы Hello World. Присутствует возможность сохранять коды своих программ без регистрации, возможность просматривать коды других пользователей, поддержание свободных руководств LibC Functions и Java API. Регистрация на сервисе предоставляет возможность создать свою библиотеку программ, а также оставлять комментарии к работам других пользователей.

После введения кода есть возможность выбора операционной системы. В случае выбора системы Windows был скомпилирован файл с расширением `exe`, в случае выбора Linux – `out`. В обоих случаях доступно скачать готовые файлы, также доступно скачать файлы с кодом в соответствующем расширении.

После ввода преднамеренной ошибки и компиляции программы сервисом было представлено сообщение компилятора с описанием ошибки.

К недостаткам сервиса можно отнести отсутствие подсветки синтаксиса, отсутствие возможности просмотра результата отработавшей программы, сравнительно более сложный интерфейс, а также присутствие рекламы. По сравнению с рассмотренными ранее сервисами, Online Compiler не содержит виджет с выводом строки HTML-кода для вставки на страницу сайта.

Обобщая результаты проведенного анализа электронных ресурсов, можно представить, каким должен быть современный сервис, поддерживающий онлайн-компиляцию:

- в поле ввода листинга сервиса должна поддерживаться подсветка синтаксиса и нумерация строк;
- сервис должен распознавать ошибки и предоставлять информацию о них;
- сервис должен обладать интерфейсом, сведенным к минимуму, также должен содержать минимальное количество рекламы;
- сервис должен поддерживать возможность скачивания файла программы в активном формате, преимуществом сервиса была бы возможность выбора соответствующего операционной системе расширения активного формата;
- сервис должен поддерживать возможность онлайн-просмотра результата отработавшей программы, что должно включать в себя поле для стандартного потока ввода данных, для реализации команд, аналогичных оператору `cin` в C++;
- наличие возможности скачать файл кода программы в соответствующем языке расширения и возможность сохранить листинг программы на сервере, для последующей его доработки;
- поддержка большего количества языков программирования даст неоспоримое преимущество электронному ресурсу;
- сервис также должен поддерживать использование различных средств компиляции для возможности экспериментирования, сравнения и анализа скомпилированных версий одной программы при помощи различных компиляторов и их версий, в дополнение будет неплохим преимуществом возможность интерпретации программ в различные языки;
- наличие теоретических материалов, таких как учебники или справочники по языкам программирования, по конкретным библиотекам, наличие готовых шаблонов и примеров готового кода было бы положительным моментом, способным превратить компилятор в образовательную систему;

– дополнительными преимуществами были бы возможность оставлять комментарии, поддержка тематических чатов, возможность просмотра кода других пользователей, наличие инструмента, предоставляющего строку для вставки демонстрации своей работы на страницу сайта.

Актуальность работы заключается в том, что реализация сформулированных требований поможет создать современный онлайн-компилятор высокого уровня, что, в свою очередь, поможет повысить эффективность самообучения начинающих программистов и уровень уже опытных программистов.

Литература

1. Лучшие онлайн компиляторы [Электронный ресурс]. – Режим доступа: <https://losst.ru/luchshie-onlajn-kompilyatory>, своб.
2. Дроздов А. Онлайн компиляторы C/C++ и не только [Электронный ресурс]. – Режим доступа: https://htrd.su/wiki/zhurnal/2013/09/19/onlajn_kompiljatory_c/c_i_ne_tolko, своб.
3. Online компиляторы [Электронный ресурс]. – Режим доступа: <http://purecodecpp.com/en/archives/3902>, своб.



Ибрагимов Дмитрий Игоревич

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107
Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: didimoner@gmail.com



Государев Илья Борисович

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, к.педагог.н., доцент
e-mail: ilia.gossoudarev@gmail.com

УДК 004

АНАЛИЗ СОВРЕМЕННЫХ СЕРВИСОВ ДЛЯ ОНЛАЙН-ВЕЩАНИЯ МУЛЬТИМЕДИЙНЫХ ПРЕЗЕНТАЦИЙ

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 мультимедиа презентации как способ удобного представления различного вида информации в доступном для зрителя виде, проводится анализ сервисов, имеющих поддержку функции онлайн-вещания презентаций.

Ключевые слова: мультимедиа, презентация, трансляция, вещание, онлайн, Microsoft Office, PowerPoint, Slidecast.

Любое выступление становится информативнее, если дополнить его мультимедийной презентацией. Презентация поможет визуализировать слова выступающего, а также показать

дополнительные данные в виде текста, изображений, воспроизведения аудио/видео роликов, демонстрации трехмерных изображений. Зритель лучше воспринимает информацию, когда ее не только рассказывает докладчик, но и показывает соответствующие ей мультимедийные материалы. Самым распространенным методом показа презентации является вывод изображения с экрана компьютера на большой экран, чаще всего при помощи цифрового видеопроектора. Этот вариант хорошо подходит для небольших аудиторий, где расстояние от зрителя до экрана позволяет в деталях разглядеть демонстрируемое изображение.

Ситуацию, когда использование экрана и проектора не обеспечивает должного качества, могло бы решить наличие у каждого зрителя (либо группы зрителей) персонального экрана, транслирующего изображение презентации. Именно о таких системах пойдет речь в данной работе. Следует заметить, что инструментов для создания и редактирования мультимедийных презентаций (как офлайн, так и онлайн) существует большое множество как платных, так и бесплатных, а систем для их онлайн-вещания разработано немного. В связи с этим **целью работы** стал поиск, обзор и анализ доступных систем для онлайн-вещания презентаций.

Перед тем как переходить к анализу систем онлайн-вещания мультимедийных презентаций, следует упомянуть роль самих презентаций в современной культуре. Лекция – традиционная форма обучения в высшей школе. Ее основная цель – создание базы для последующего изучения студентами учебного материала. Необходимо отметить, что традиционная (классическая) лекция обладает рядом существенных недостатков. И, прежде всего, традиционная лекция – это в основном монологический способ изложения материала, в такой ситуации студенты – пассивные слушатели. Нередко представляемый лектором на доске иллюстративный материал выходит запутанным и некачественным [1].

Мультимедиа презентация – это уникальный и самый современный на сегодняшний день способ представления информации. Информативность электронных презентаций намного выше традиционных выступлений за счет мультимедийности – сочетания в презентации нескольких типов информации, таких как текст, графика, анимация, видео и звук, что дает возможность одновременного воздействия сразу на несколько каналов восприятия ребенка. Известно, что человек большую часть информации воспринимает органами зрения (~80%), и органами слуха (~15%) (это давно замечено и эффективно используется). Мультимедиа технологии позволяют воздействовать одновременно на эти важнейшие органы чувств человека [2]. В отличие от видео, мультимедиа технологии позволяют управлять потоком информации, т.е. могут быть интерактивны. Мультимедиа презентации дают прямой доступ к информации. Пользователь может сразу видеть все содержание и переходить к тому, что его заинтересовало [3]. Особое место в мультимедийных презентациях занимают системы онлайн-вещания [4]. Основной задачей такой системы является предоставление клиенту актуальных данных о презентации в реальном времени. Другими словами, человек, подключившийся к презентации докладчика, видит на дисплее своего устройства саму презентацию, а также все манипуляции, которые над ней проводит докладчик.

Для сравнительного анализа были выбраны следующие сервисы, позволяющие проводить онлайн-трансляцию презентаций: Microsoft Office PowerPoint и Slidescast.

Microsoft Office – это пакет офисных программ для операционной системы Windows, в состав которого входит приложение для работы с презентациями PowerPoint. Данное приложение широко распространено, а формат презентаций .pptx поддерживается практически всеми инструментами для работы с ними. Сам продукт представляет собой мощный редактор с богатым набором инструментов для создания мультимедийных презентаций различной сложности, поддерживающий большинство популярных форматов контента (текст, изображение, звук, видео, анимация). Средствами этого же приложения можно устроить показ созданной презентации, при этом изображение будет выводиться на экран компьютера. Однако имеется и функция для трансляции презентации через сеть

Интернет. Для этого в настольном приложении MS Office PowerPoint загружается необходимая презентация, затем устанавливается соединение с сервером Microsoft, на который загружается активная презентация. Далее владельцу презентации выдается url-адрес, который необходимо предоставить клиентам. MS Office PowerPoint предлагает длинный и трудночитаемый url (пример: k1-broadcast.officeapps.live.com/m/Broadcast.aspx?Fi=8890b9b89b22c31c%5Fd4b07ecd%2D17ae%2D4687%2D8882%2D0436a4062c03%2Epptx). После перехода по данному адресу, в интернет-браузере загружается онлайн-версия приложения PowerPoint, в котором отображается актуальное состояние презентации, а также происходит постоянное обновление информации в фоновом режиме.

Slidecast [slidecast.ru] – это бесплатный веб-сервис, представляющий собой платформу для создания, хранения и демонстрации мультимедийных презентаций. Для того чтобы получить доступ к инструментам платформы, необходимо зарегистрировать учетную запись на сервере slidecast, затем пользователь может воспользоваться личным хранилищем презентаций.

Добавить презентации можно онлайн, в самой системе, вызвав удобный редактор презентаций, а также есть возможность загрузить готовую презентацию в формате .pdf. После того, как работа над мультимедийной презентацией завершена, можно приступить к ее трансляции. Система устроена так, что окно со слайдами отображается независимо от элементов управления, называемых «кликером». Доступ к «кликеру» можно получить как с компьютера, так и со смартфона или планшета, поэтому управлять показом становится намного удобнее. Для того чтобы зрители могли подключиться к транслируемой презентации, им необходимо перейти по короткой удобочитаемой ссылке, которую генерирует сервис для каждой хранимой в личном кабинете пользователя презентации (пример: sld.pt/wnv).

Помимо функции трансляции мультимедийных презентаций, slidecast обладает возможностью устраивать интерактивные опросы с последующим выводом статистики, а также задавать вопросы, на которые каждый зритель отвечает на собственном устройстве, подключенном к показу (рисунок).

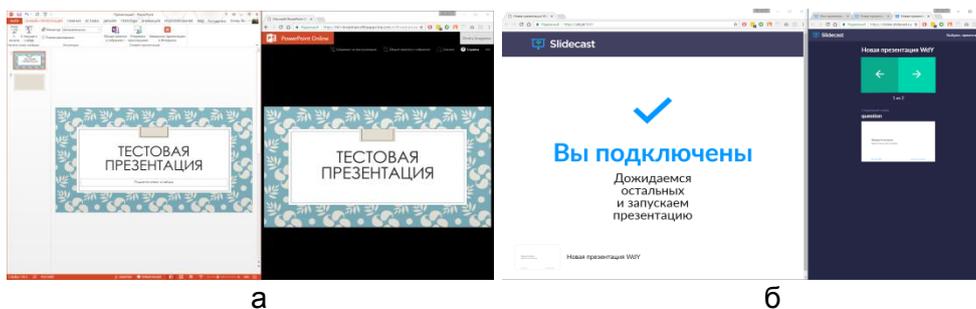


Рисунок. Окно управления презентацией и окно просмотра, MS Office PowerPoint (а) и Slidecast (б)

Следует сказать, что в обеих системах требуется постоянное стабильное подключение к сети Интернет, а все данные отправляются на серверы приложений. Для тестирования быстродействия была создана презентация, состоящая из пяти одинаковых слайдов. Среднее время загрузки презентации в Slidecast составило 4–5 с, а время загрузки в PowerPoint – 10–12 с. При тестировании использовался Интернет с каналом в 40 Мбит, среднее значение ping 90 мс).

В заключение следует отметить, что сервисы для трансляции презентаций не очень широко распространены в настоящее время, поэтому данная область является актуальной для проведения исследований и проектирования системы, решающей упомянутую задачу. Анализ, проведенный в рамках данной работы, помог выявить сильные стороны систем, которые следует учесть при разработке подобного рода сервиса, а также слабые стороны, на которые следует обратить внимание и попытаться исправить или избежать [5].

Литература

1. Талипова И.П., Феоктистова Л.А., Рзаева Т.В. К вопросу о чтении лекций с использованием презентаций MS Powerpoint // Сб. ст. Социально-экономические и технические системы: исследование, проектирование, оптимизация. – 2016. – № 3(70). – С. 65–72.
2. Ванюкова Е.С. Использование мультимедиа презентаций в учебном процессе // Сб. ст. Вестник научных конференций. – 2015. – № 3-5(3). – С. 29–30.
3. Сальникова А.О., Городищева А.Н. Компьютерные презентации с использованием мультимедиа технологий // Сб. ст. Актуальные проблемы авиации и космонавтики. – 2012. – Т. 2 – № 8. – С. 316–317.
4. Государев И.Б. Развертывание и интеграция инновационных учебных сред: бордкастинг, облачные хостинги и edX // Компьютерные инструменты в образовании. – 2014. – № 1. – С. 26–35.
5. Афанасьев А.А., Грушевская В.Ю. Сравнительный анализ онлайн-сервисов для создания презентаций // Сб. ст. Актуальные вопросы преподавания математики, информатики и информационных технологий. – 2016. – С. 128–136.

**Изукаев Егор Валерьевич**

Год рождения: 1992

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4207Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: egor.izukaev@gmail.com

УДК 004.438

**ИССЛЕДОВАНИЕ ВНЕДРЕНИЯ ТЕХНОЛОГИЙ ПРЕПРОЦЕССИНГА
НА ВЕБ-ФРОНТЕНДЕ****Изукаев Е.В.****Научный руководитель – к.педагог.н., доцент Государев И.Б.**

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе рассмотрены препроцессоры CSS. Описаны основные преимущества использования препроцессоров.

Ключевые слова: CSS, препроцессор, Less, Stylus, SASS.

В настоящее время архитектура веб-интерфейсов становится все сложнее, разработка веб-интерфейсов занимает все больше как временных, так и трудовых ресурсов. Для решения данных проблем разработчикам все чаще приходится использовать дополнительные библиотеки, надстройки, фреймворки.

Основной составляющей в создании веб-страниц являются стили. К ним можно отнести следующий ряд проблем:

- дублирование стилей;
- в синтаксисе CSS отсутствует понятия вложенности, что ухудшает визуальное восприятие;

– возможные проблемы с поддержкой стилей проекта на протяжении всего жизненного цикла.

Все это приводит разработчиков на мысль об оптимизации данного рабочего процесса.

Для решения данных проблем существуют CSS-препроцессоры, которые облегчают работу со стилями за счет того, что препроцессоры добавляют новые возможности, такие как: использование переменных, примесей, функций, массивов, циклов, которые не доступны в обычном CSS.

Целью работы являлось исследование внедрения технологий препроцессинга, выделение преимуществ и недостатков их использования.

CSS-препроцессоры называются стилевыми языками с дополнительными возможностями, которые отсутствуют в CSS.

Основная идея появления препроцессоров заключалась в предоставлении удобных синтаксических конструкций с целью упростить разработку, а также поддержку проекта в будущем [1].

CSS-препроцессоры дают ряд преимуществ в сравнении с CSS:

- визуальное восприятие;
- структурированность;
- логичность;
- упрощение кроссбраузерной верстки.

В настоящее время наиболее популярными препроцессорами являются: SCSS (Sass), Less, Stylus. Основным отличием данных препроцессоров является синтаксис.

Синтаксис в SCSS и Stylus более разнообразней в сравнении с Less. Данные препроцессоры предоставляют возможность использовать стандартный синтаксис CSS и синтаксис с необязательной расстановкой двоеточия, скобок и точек с запятой. В препроцессоре Stylus переменные могут быть объявлены различным способом в отличие от SCSS, где переменные всегда начинаются с символа «&» и Less, где переменные начинаются с символа «@» [2]. В синтаксисе Stylus также в отличие от стальных препроцессоров при объявлении переменных не ставится точка с запятой, а присвоение значений происходит с помощью знака равенства.

Одним из главных преимуществ использования технологий препроцессинга является построение каскада. В CSS присутствует логическое каскадирование, но вложение одного блока стилей в другой невозможен в отличие от препроцессоров [2].

Также к преимуществам использования препроцессоров можно отнести примеси. Примеси предоставляют возможность многократно использовать CSS-свойства, объявив их один раз внутри примеси. Они могут быть полезны для упрощения написания кода с целью поддержки кроссбраузерности с помощью автоматизации расстановки префиксов.

Наследование в препроцессорах играет важную роль и помогает разработчику избежать написания дополнительных классов HTML-тегам. В препроцессорах SCSS, Stylus применяется директива «@extend», которая предоставляет возможность наследования свойств CSS от одного селектора к другому. Синтаксис Less не предоставляет возможности наследования в таком виде, но наследование можно осуществить с помощью примесей [2].

Также препроцессоры предлагают возможность применять математические операции.

Сравнительный анализ использования технологий препроцессинга позволяет сделать следующие выводы:

- препроцессоры CSS, в сравнении с привычным нам видом CSS, предлагают синтаксис с использованием вложенности улучшающий визуальное восприятие кода;
- внедрение технологий препроцессинга позволяет сократить написание кода за счет ранее недоступных возможностей в CSS;
- так как препроцессоры являются надстройкой над стандартным языком стилей CSS, они имеют свой собственный синтаксис, который браузеры не воспринимают;
- модификация стандартного CSS невозможна.

Литература

1. CSS препроцессоры [Электронный ресурс]. – Режим доступа: https://mrmlnc.gitbooks.io/less-guidebook-for-beginners/content/chapter_1/preprocessors, своб.
2. Sass vs Less vs Stylus [Электронный ресурс]. – Режим доступа: <https://code.tutsplus.com/tutorials/sass-vs-less-vs-stylus-preprocessor-shootout--net-24320>, своб.



Казначеева Екатерина Олеговна

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

email: katya.kazn@gmail.com



Государев Илья Борисович

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна,

к.педагог.н., доцент

e-mail: ilia.gossoudarev@gmail.com

УДК 004

ЭВОЛЮЦИЯ ФОРМАТОВ ОБМЕНА ДАННЫМИ НА ВЕБ-ПЛАТФОРМЕ НА ПРИМЕРЕ XML И JSON

Казначеева Е.О.

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 наиболее популярных формата обмена данными XML и JSON для клиент-серверных приложений. Приведены определения и описания каждого из форматов, а также результаты анализа на основе личного опыта автора данной работы. Выделены общие черты и различия двух представленных форматов, что позволяет сделать вывод о том, какую из технологий в настоящее время можно рекомендовать большинству разработчиков программного обеспечения.

Ключевые слова: XML, JSON, веб-приложение, формат обмена данными, обмен данными.

Стремительные темпы развития информационных технологий уже давно стали нормой в современном мире. В частности, веб-разработка тоже не стоит на месте. Новые технологии для создания веб-приложений направлены на улучшение всех компонентов разработки, например, удобство для пользователя, быстрота разработки, надежность работы, эффективное использование ресурсов и другие. Создание веб-приложения подразумевает взаимодействие клиентской и серверной сторон, а именно обмен данными между ними. Единой технологии для данного процесса нет, но основными и самыми распространенными форматами являются XML и JSON, каждая из которых обладает своими достоинствами и недостатками. При всем этом возникает проблема выбора той или иной технологии. Решение данной проблемы может значительно упростить дальнейшую работу при разработке приложения, а также минимизировать порождение некоторых других проблем.

XML (eXtensible Markup Language) – расширяемый язык разметки документов, который является подмножеством метаязыка SGML (Standard Generalized Markup Language – стандартный обобщенный язык разметки). Он был создан для структурирования, хранения и передачи информации. Первое упоминание об XML было в 1996 году, тогда и появилась начальная рабочая версия данного формата, а уже в 1998 году международная организация W3C утвердила спецификацию «Extensible Markup Language (XML) 1.0». Этот язык используется в качестве средства для описания грамматики других языков и контроля за правильностью составления документов [1]. Это означает, что тэги XML не заданы с самого начала и разработчику предоставляется право определить все тэги и их порядок самостоятельно. Обработка XML-документа обычно происходит с использованием таких технологий как DTD, XDR, XSD-схем, XPath, XSLT и других [2]. Он дал начало таким популярным на сегодняшний день форматам, как XHTML, SVG, RSS и др.

JSON (JavaScript Object Notation) – формат обмена данными, основанный на подмножестве языка программирования JavaScript [3], который на данный момент является очень популярным языком для создания интерактивности на веб-странице. При этом JSON не зависит от JavaScript, и может быть использован практически с любым языком программирования. Формат JSON был определен в стандарте ECMAScript в декабре 1999 года. На данный момент, практически все JavaScript-программисты выбирают именно этот формат для передачи по сети объекта данных от сервера к клиенту.

Для того чтобы выяснить, какой формат обмена данных подойдет для того или иного приложения, необходимо их сравнить. Дальнейшее сравнение основывается только на опыте и мнении автора данной работы.

Для начала создадим два документа, которые будут определять один и тот же объект, а именно массив, состоящий из трех сотрудников определенной компании [4].

Пример на JSON:

```
{ "employees": [
  { "firstName": "John", "lastName": "Doe" },
  { "firstName": "Anna", "lastName": "Smith" },
  { "firstName": "Peter", "lastName": "Jones" }
]}
```

Пример на XML:

```
<employees>
<employee>
<firstName>John</firstName>
<lastName>Doe</lastName>
</employee>
<employee>
<firstName>Anna</firstName>
<lastName>Smith</lastName>
</employee>
```

```
<employee>
<firstName>Peter</firstName>
<lastName>Jones</lastName>
</employee>
</employees>
```

В ходе исследования был проведен анализ каждого из двух представленных форматов обмена данными, после чего были сделаны следующие наблюдения:

1. оба формата являются удобочитаемыми. Безусловно, визуальное восприятие кода очень важный фактор, но он довольно субъективен, так, многие специалисты ведут дискуссию по поводу того, является ли удобочитаемость формата обмена данными самым главным пунктом при выборе формата или же данный момент не столь существенный [5]. Однако следует заметить, что JSON компактнее в написании, чем XML, так как не имеет закрывающих тэгов. В свою очередь, XML будет более понятен для начинающего разработчика, но JSON удобнее для быстрого прочтения описываемой предметной области;
2. оба формата могут быть интерпретированы с помощью большого количества языков программирования. Парсинг каждого из сравниваемых форматов может быть осуществлен с помощью различных языков программирования [3], например, Python, PHP, C++, Java, JavaScript и других. Данный фактор непременно очень важен, так как практически любой программист может их использовать и не искать другую технологию, которая бы подошла для используемого ими языка программирования;
3. оба формата могут работать с XMLHttpRequest. Объект XMLHttpRequest, который дает возможность браузеру делать HTTP-запросы к серверу без перезагрузки страницы, может работать с данными в любом текстовом формате, в том числе и с XML и JSON. Это важно, так как на сегодняшний день все более и более часто данная технология используется при создании сайтов;
4. JSON может использовать массивы. Работа с массивами является важным свойством формата обмена данными, так как нередко передаваемая информация состоит из большого количества повторяющихся структур, а создание подобного объекта в XML будет довольно трудоемким процессом;
5. парсинг JSON удобнее и быстрее, чем парсинг XML. Действительно, для того чтобы получить данные из XML используя JavaScript, нужно: определить XML-документ, использовать DOMParser() для создания парсера, затем с помощью парсера и метода parseFromString() создать XMLDOM-объект, после чего можно определять новые переменные и присваивать им значения. В случае с JSON нужно всего лишь использовать функцию JSON.parse(), после чего данные становятся JavaScript-объектом, и их уже можно использовать для последующих задач.

Подводя итоги, следует сказать, что оба формата довольно широко распространены, и поэтому поддерживаются многими редакторами, а также каждый из них нашел огромное количество применений. XML и JSON имеют как общие черты, так и отличия, некоторые из них были изложены в данной работе. На данный момент формат JSON пользуется наибольшей популярностью среди сообщества разработчиков программного обеспечения. Из-за своей простоты при создании и обработке он привлек внимание многих специалистов в данной области. Все чаще при разработке веб-приложений можно наблюдать, что при выборе формата обмена данными XML уступает JSON.

Литература

1. Печерский А. Язык XML – практическое введение [Электронный ресурс]. – Режим доступа: <http://www.codenet.ru/webmast/xml/>, своб.
2. Одиночкина С.В. Основы технологий XML. – СПб.: НИУ ИТМО, 2013. – 56 с.

3. Введение в JSON [Электронный ресурс]. – Режим доступа: <http://www.json.org/json-ru.html>, своб.
4. JSONvsXML [Электронный ресурс]. – Режим доступа: http://www.w3schools.com/js/js_json_xml.asp, своб.
5. Канаев К.А., Фалеева Е.В., Пономарчук Ю.В. Сравнительный анализ форматов обмена данными, используемых в приложениях с клиент-серверной архитектурой // Фундаментальные исследования. – 2015. – № 2-25. – С. 5569–5572.



Калашников Дмитрий Андреевич

Год рождения: 1994

Факультет информационных технологий и программирования,
кафедра информационных систем, группа № М4205

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: kalashnikovdm@gmail.com

УДК 004.422.81

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА СИСТЕМЫ ДЛЯ АВТОМАТИЗИРОВАННОГО СБОРА И ОБРАБОТКИ ДАННЫХ ИЗ ВНЕШНИХ ИСТОЧНИКОВ ДАННЫХ

Калашников Д.А.

Научный руководитель – тьютор Хвастунов А.П.

В работе рассмотрена расширяемая, горизонтально масштабируемая и распределенная архитектура и элементы реализации системы, ориентированной на сбор данных из гетерогенных источников (веб-сайты, внешние программные интерфейсы) и их последующую обработку (подсчет агрегированных метрик, преобразование формата). По результатам внедрения системы команда маркетинговых аналитиков смогла автоматизировать получение и обработку данных о вакансиях и репозиториях с открытым исходным кодом, тем самым значительно сократив время проведения типовых исследований.

Ключевые слова: распределенная система, обработка данных, сбор данных, масштабируемость, Docker, Consul, Spring, MongoDB, расширяемая система.

В работе рассмотрена система, которая проектировалась и разрабатывалась на основании потребностей команды аналитиков для выполнения типовых задач, связанных с оценкой рынка программистов. Автоматизация маркетинговых исследований является важным аспектом улучшения качества услуг, предоставляемых отделом [1]. Она позволяет:

- ускорить проведение типовых исследований;
- освободить аналитиков от обязанностей по подготовке инструментов для сбора и пост-обработки данных и их окружения;
- избавить от ручного отслеживания хода исполнения инструментов;
- создать единую унифицированную систему для реализации последующих решений для сбора или обработки данных;
- предоставить единое место для хранения всех данных.

Источниками данных для системы являются:

- веб-сайты;
- репозитории с открытым исходным кодом;
- внешние программные интерфейсы;
- внутренние системы статистики.

Несмотря на наличие различных готовых решений для сбора данных с веб-сайтов (например, Import.io [2]), после апробации было принято решение об отказе от их использования. Это связано с тем, что часть из них не подходит по экономическим соображениям, а другие – не предоставляют всей функциональности, требуемой сбора данных (выгрузка файлов, переход по страницам исполнением собственных JavaScript-файлов). В ходе анализа существующих решений не было обнаружено ни одного сервиса или приложения, решающего задачи для сбора данных из репозитория с открытым исходным кодом.

На начальном этапе было принято решение о разработке системы распределенного типа, построенной по принципу микросервисов для обеспечения расширяемости и масштабируемости.

Далее был произведен анализ технологий, которые могут быть использованы для создания такой системы. По результатам анализа были выбраны следующие технологии:

- Consul [3] – является сервисом для обнаружения и конфигурации сервисов. Он предоставляет встроенное хранилище по типу ключ-значение, механизмы для проверки состояния работоспособности сервисов и возможности для построения распределенных блокировок. Consul состоит из двух частей – серверов и клиентов. Первые хранят у себя всю информацию о системе и производят ее репликацию. Вторые являются конечными точками, с которыми взаимодействуют все компоненты в системе;
- MongoDB (документно-ориентированная нереляционная база данных) – выбрана для хранения результатов сбора данных. Одной из главных причин такого выбора является тот факт, что невозможно выделить единообразную структуру данных из-за того, что она полностью зависит от конфигурационного файла обработчиков;
- Docker и Docker Swarm – предоставляют механизмы для контейнеризации всех сервисов и единообразного запуска контейнеров на кластере виртуальных машин. Контейнеризация процессов позволяет четко разделить окружение каждого из них;
- PostgreSQL – это реляционная база данных с открытым исходным кодом. В рамках системы ее основной задачей является хранение сопутствующей информации для планировщика и третьесторонних сервисов;
- Каркас Spring – это большая коллекция различных более мелких каркасов для Java [4]: реализация паттерна MVC, упрощение работы с базами данных, развертывание веб-сервиса внутри встроенного сервера приложений и т.д. Spring предоставляет модуль для работы с Consul в терминах своего базового каркаса. Он позволяет автоматически регистрировать сервис в Consul при старте и автоконфигурирует стандартные конечные точки для проверки работоспособности. Spring также предоставляет модули для упрощенной работы с GridFS и MongoDB, позволяя получать информацию о местоположении базы данных через Consul.

На основании выбранных технологий была разработана следующая архитектура, представленная на рисунке. На данной диаграмме все компоненты системы представляют собой docker-контейнеры, которые запущены на нескольких виртуальных машинах, объединенных в кластер с помощью Docker Swarm.

Система состоит из следующих компонентов:

- клиент – предоставляет интерфейсы для запуска обработчиков внутри системы и отслеживания их состояния;
- компонент – для запуска контейнеров предназначен для запуска других компонентов в виде docker-контейнеров в системе;

- обработчик – выполняет задачи по сбору или обработке данных. Он может быть запущен только через взаимодействие с компонентом Клиент и может взаимодействовать с любыми сервисами внутри системы, а также производить масштабирование сервисов через Компонент для запуска контейнеров;
- сервис – является долгоживущей сущностью внутри системы, с которой могут взаимодействовать компоненты Обработчик или иные сервисы;
- планировщик запускает задачи по расписанию или по триггерам, основанным на результатах исполнения компонентов типа Обработчик.

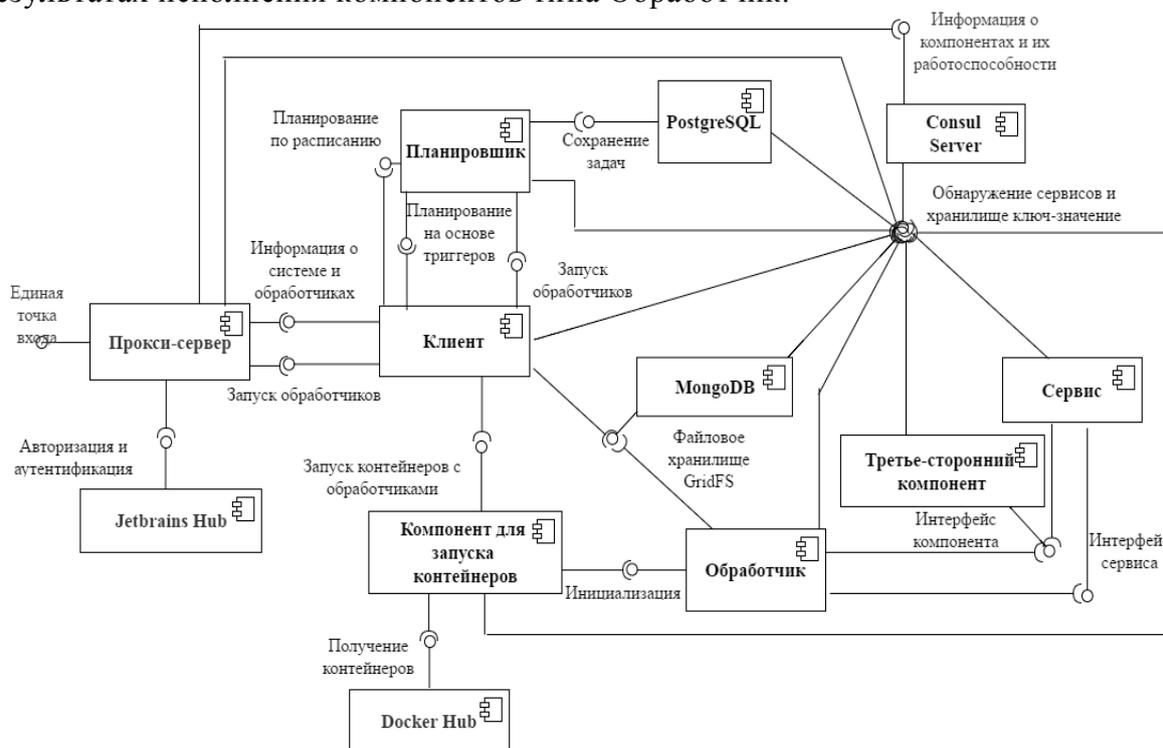


Рисунок. Обобщенная архитектура системы

В рамках разрабатываемой системы имеются следующие сложные вопросы интеграции:

- динамическое обнаружение компонентами системы других компонентов, сетевые адреса которых могут быть присвоены автоматически и чье физическое местоположение не задано строго;
- быстрое обнаружение отказа определенного компонента, распространение этой информации и перенаправление запросов на аналогичные работающие компоненты;
- предотвращение цепочки каскадных отказов;
- оповещение одним компонентом всех компонентов внутри системы об изменении своего состояния, обеспечение условного ожидания одного компонента на изменение состояния другого.

В исследовании был произведен анализ и апробация собственных решений и третьесторонних библиотек, систем и каркасов, позволяющих полностью или частично разрешить возникшие вопросы. Основной системой, гарантирующей динамическую регистрацию сервисов, мониторинг состояния «здоровья» и отслеживание изменений состояния обработчиков, является Consul. За счет его архитектуры также гарантируется его отказоустойчивость, масштабируемость и корректное распространение требуемой информации внутри системы. Библиотека Hystrix предоставляет шаблон «Прерыватель» [5], защищающий от возникновения каскадных отказов при совершении операций, требующих взаимодействия между собой нескольких компонентов. Единообразная работа с данными элементами гарантируется путем использования каркаса SpringCloud, который также

предоставляет дополнительные интерфейсы, упрощающие работу с библиотеками и системами.

В работе были рассмотрены принципы построения архитектуры компонентов и сложные моменты их интеграции для системы автоматизированного сбора данных из внешних источников. Благодаря отобранным к использованию сторонним системам и каркасам, при разработке системы удалось не только значительно минимизировать объемы собственного кода, но и разрешить многие проблемы интеграции, обеспечить масштабируемость и расширяемость системы. После введения системы в эксплуатацию было значительно сокращено время проведения исследований, а их процесс был сильно упрощен для аналитиков, не имеющих опыта в программировании.

Литература

1. Kasabov E., Warlow A. Automated Market and Marketing Research // The Compliance Business and Its Customers: Gaining Competitive Advantage by Controlling Your Customers. – Palgrave Macmillan UK – London, 2001. – P. 60–72.
2. Import.io [Электронный ресурс]. – Режим доступа: <https://www.import.io/>, своб.
3. Consul.io [Электронный ресурс]. – Режим доступа: https://www.consul.io, своб.
4. Spring Projects [Электронный ресурс]. – Режим доступа: <https://spring.io/projects>, своб.
5. Nygard M.T. Release It! Design and Deploy Production Ready Software. – The Pragmatic Bookshelf, 2007. – P. 93–96.



Калинина Кристина Владимировна

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: tiinaaaliiv@gmail.com

УДК 004.92

ВЫЯВЛЕНИЕ НАЧАЛЬНЫХ ОШИБОК ПРИ СОСТАВЛЕНИИ ЗАДАНИЯ ДИЗАЙН-ПРОЕКТА

Калинина К.В.

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

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

Работа посвящена анализу взаимодействия заказчика и дизайнера. Описаны основные этапы работы создания дизайн-проекта. Рассмотрены основные ошибки, совершаемые на начальных этапах работы. Описаны основные принципы взаимодействия заказчика и дизайнера во время работы над проектом.

Ключевые слова: взаимодействие заказчика и дизайнера, создание дизайн-проекта, этапы работы при создании дизайн-проекта, ошибки взаимодействия заказчика и дизайнера, принципы взаимодействия заказчика и дизайнера.

Взаимодействие заказчика и дизайнера при разработке дизайн-макета является не простой задачей. Во время этой работы нередко возникают ошибки создания технического задания для выполнения. Для того чтобы их избежать, необходимо определить основные этапы работы над проектом и основные ошибки.

Дизайн – это особая сфера деятельности. В ней не существует определенной системы оценивания готового проекта, а также одна из немногих сфер, где результат не сразу согласуется с заказчиком, а проходит несколько этапов. Современный дизайн делится на четыре основных вида: графический дизайн, промышленный дизайн, архитектурный дизайн и дизайн среды. В каждой из этих сфер есть конкретные субъекты и объекты проектирования, функции и формы [1]. Несмотря на различия этих видов дизайна, для них можно выделить основные этапы работы при создании конечного продукта.

Можно выделить следующие основные этапы работы при разработке дизайн-макета:

1. проведение переговоров и составление технического задания, или брифа;
2. исследование требуемой сферы деятельности заказчика, анализ конкурентов и поиск аналогов;
3. создание эскизов проекта;
4. обсуждение с заказчиком всех вариантов эскизов проекта, внесение изменений и доработок;
5. проектирование и внесение последних изменений и корректировок в дизайн-макет;
6. адаптация готового дизайн-макета к основным элементам компании заказчика [2].

Этап переговоров и составления технического задания наиболее важен для дальнейшей работы. Именно на этом этапе определяется то, что хочет заказчик, и какой вид будет иметь конечный проект. В этой связи необходимо грамотно подходить к составлению технического задания, ведь оно определяет всю дальнейшую работу над проектом. Исследование сферы деятельности заказчика также играет важную роль в проектировании будущего дизайн-макета. Дизайнер должен понимать, в какой области работает заказчик и что имеет большое значение для этой области деятельности.

При взаимодействии заказчика и дизайнера на первых этапах можно выделить следующие ошибки:

- дизайнер не ставит себя на место заказчика и не пытается понять важные аспекты его деятельности;
- при неясных моментах, дизайнер часто додумывает решение сам, а не уточняет у заказчика;
- заказчик хочет видеть в одном проекте стили, которые не совместимы;
- дизайнер общается с заказчиком с помощью специфических терминов [3];
- при планировании проекта не учитываются ограничения для выпуска готового решения;
- заказчик часто слушает советы других людей (знакомых, родственников, коллег) и проект, который задумывался в начале, приобретает другой вид;
- работа над проектом не делится на этапы и не структурируется, или не определяются цели для каждого этапа;
- дизайнер и заказчик не понимают друг друга при обсуждении вопросов [4].

Любые из этих ошибок значительно усложняют работу над проектом и в конечном итоге дают отрицательное влияние на конечный результат работы.

Можно выделить основные принципы работы над проектом и взаимодействия заказчика и дизайнера.

1. Нужно получать наиболее детальное представление о желаниях заказчика и его сфере деятельности. Для получения информации могут использоваться книги, статьи, интернет-источники и сам заказчик [2].
2. Для больших проектов необходим человек, который будет управлять процессом и с творческой стороны, и со стороны заказчика. Иначе работа будет не организована и появится гораздо большее количество ошибок, не связанных с творческой работой дизайнера. Обычно в таком случае приглашают менеджера-консультанта с дизайнерским образованием, который несет большую ответственность по управлению проектом и взаимодействию.

3. Все важные вопросы и задачи нужно задокументировать в техническом задании или брифе. Также, по возможности, вести запись всех личных и телефонных переговоров. Записи помогут работать в нужном направлении при создании проекта.
4. При сборе информации, необходимо получить как можно больше сведений об организации заказчика. К таким данным могут относиться: название организации, ее расположение, сколько лет существует, количество сотрудников, какие товары или услуги предоставляет, миссия компании, конкуренты и т.д.
5. При составлении брифа, нужно задавать правильные вопросы.
6. Необходимо вести регламент обсуждений, заранее определить встречи с заказчиком и затрачиваемое на них время. Так, заказчик сможет за отведенное количество времени составить комплекс вопросов, которые необходимо решить в его проекте [5].
7. Не существует оценки, которая может точно охарактеризовать созданный эскиз или макет. Все такие оценки основываются на личном мнении.
8. Необходимо всегда учитывать ограничения проекта. Ограничения могут возникать как на начальном этапе, так и при реализации готового дизайн-проекта.
9. Нужно рассматривать примеры уже созданных работ, чтобы лучше понимать, что нравится заказчику [3].

Проведенный анализ будет использован для оптимизации взаимодействия заказчика и исполнителя при разработке дизайн-проекта.

Литература

1. Минервин Г.Б., Шимко В.Т. Ефимов А.В. Дизайн. Иллюстрированный словарь-справочник. – М.: Архитектура-С, 2004. – 288 с.
2. Марченко М.Н. Федосеева А.А. Процесс проектирования фирменного стиля и взаимодействие дизайнера с потребителем его услуг // Наука, образование, общество. – 2015. – № 2(4). – С. 58–61.
3. Эйри Д. Логотип и фирменный стиль. Руководство дизайнера. – СПб.: Питер, 2011. – 208 с.
4. Весёлкина М.В. Творческий диалог дизайнера с потенциальным заказчиком // Научный альманах. – Тамбов, 2016. – № 3-4. – С. 365–367.
5. Весёлкина М.В. Аспекты эффективных переговоров в сфере дизайна // Инновации в науке: сб. ст. по матер. LVI междунар. науч.-практ. конф. – 2016. – № 4(53). – Ч. I. – С. 50–54.



Каллавуc Мари Меэлисовна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4106

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: marie.kallavus@gmail.com

УДК 004.414.32

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ИНСТРУМЕНТОВ ПРОТОТИПИРОВАНИЯ ИНТЕРФЕЙСА ИНЖЕНЕРНЫХ СИСТЕМ

Каллавуc М.М.

Научный руководитель – д.т.н., профессор Погорелов В.И.

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе предложен метод прототипирования, включающий в себя комбинирование инструментов для создания макетов в виде графических инструментов, блок-схем и десктопных приложений для разработки прототипов. Приведены преимущества использования данного метода для разработки прототипов интерфейса инженерных систем.

Ключевые слова: прототипирование, инженерные системы, интерфейс.

Инженерные системы – это программные комплексы, предназначенные для решения инженерных целей и задач [1].

Инженерные системы разрабатываются научно-инженерными центрами, исследовательскими институтами и прочими предприятиями.

Свойства инженерных систем следующие [2]:

- большой объем входных данных;
- большой объем выходных данных;
- множество параметров для расчетов;
- использование математических формул и моделей;
- использование различных режимов для обработки информации;
- выведение и (или) сохранение графиков.

В данной работе предлагаются следующие методы прототипирования:

- графические программные средства;
- динамические программные средства;
- блок-схемные программные средства;
- десктопные программные средства;
- онлайн программные средства.

Можно утверждать, что для разработки именно инженерных систем такие односоставные методы, как графический, блок-схемный, динамический методы и метод с использованием онлайн-инструментов для прототипирования, не подходят, поскольку не удовлетворяют критериям разработки инженерных систем.

Однако можно выявить указания по использованию данных методов. Графический метод подойдет для дизайнеров для разработки логотипов, иллюстраций, а также макетов веб-страниц и лэндинг пэйджей.

Блок-схемный метод удобен для разработчиков, поскольку блок-схемы составляют для них план работ, следуя порядку которых можно изготовить финальный продукт.

Динамический метод удобен для создания презентаций и словесного описания предполагаемой разрабатываемой системы заказчику, а также для демонстрации уже разработанной системы с помощью презентации.

Метод с использованием онлайн-инструментов для прототипирования может быть рекомендован небольшим конторам, поскольку является простым для доступа, эффективным и удобным для демонстрации заказчику и для тестирования реальными клиентами.

Метод с использованием десктопных программ удовлетворяет многим критериям, являющихся ключевыми для разработки метода прототипирования интерфейса инженерных систем. Однако данный метод имеет собственные недостатки, например, минусом является невысокая степень проработки структуры проекта, что в дальнейшем может привести к ошибкам работы системы, а также невысокая степень «дизайнерского» уровня макеты [2].

Для устранения ошибок данного метода в работе предложены следующие комбинированные методы прототипирования:

- сочетание блок-схемных и десктопных программ;
- сочетание графических и десктопных программ;
- сочетание графических, блок-схемных и десктопных программ.

В результате тщательного анализа разработанных методов и выделения их преимуществ и недостатков можно сделать вывод, что именно комбинированный метод,

сочетающий графические, блок-схемные и десктопные программ, является наиболее предпочтительным для создания прототипов инженерных систем.

Данный метод включает в себя все достоинства трех методов, которые взаимно сводят к минимуму большинство недостатков каждого из методов по отдельности.

С помощью данного метода появляется возможность:

- протестировать разрабатываемый прототип на нескольких промежуточных этапах во время этапа проектирования;
- тщательно продумать логику структуры проекта [3];
- подготовить нарисованные графические элементы для дальнейшего использования техническими писателями;
- подготовить нарисованные графические элементы для передачи на следующий этап кодовой разработки системы программистами;
- создать интерактивный прототип, который является как макетом для демонстрации заказчику, так и действующим наглядным ТЗ для программистов [4, 5];
- сделать разрабатываемый прототип не только логически продуманным и интерактивным, но и наглядным с помощью графических элементов.

Также можно сказать, что метод, сочетающий блок-схемные и десктопные программы, подойдет для создания строгих обезличенных программ с минималистическим интерфейсом, а также для экономии времени на этапе прототипирования. Данный метод объединяет достоинства блок-схемного метода и метода с использованием десктопных программ и сводит к минимуму количество логических ошибок, которые появились бы вследствие использования любого односоставного метода.

Метод, сочетающий графические и десктопные программы, объединяет в себе достоинства двух методов по отдельности, однако недостатком данного комбинированного метода является невысокая степень проработки структуры проекта. Данный метод подойдет для более быстрого процесса создания интерактивного и наглядного прототипа. И, если свести взаимодействие заказчика с прототипом к минимуму (в ходе которого заказчик может выявить логические неточности или просто сделать вывод о том, что его не устраивает какая-нибудь последовательность процедур в системе, что означает, что придется переделывать прототип), то данный метод также пригоден к использованию.

Для проверки эффективности разработанного метода использован метод опроса. Проанализировав результаты, можно сделать следующие выводы:

- большинство опрошенных имеют опыт работы со сложными системами, причем многие указывают на неудобство дизайна систем, а также на невысокую степень логичности переходов между страницами систем. Таким образом, можно сказать, что необходима высокая степень проработки структуры каждого проекта, особенно многостраничных и многофункциональных технических систем;
- большинство опрошиваемых признали, что интерактивный прототип был бы более удобен для тестирования, просмотра или пользования. Таким образом, можно сказать, что доказано преимущество использования десктопных программ для создания интерактивных прототипов не только для разработчиков, но и для потенциальных заказчиков или тестировщиков – согласно результатам опроса 87,5% опрошенных было бы удобнее, если бы макет откликался на их действия;
- практически все опрошенные ответили, что прототип с детальной прорисовкой кажется им законченным и завершенным, а также наглядно демонстрирующим потенциальные возможности и функциональность системы, в то время, как прототип без детальной прорисовки (т.е. без графических элементов) кажется им сырым и незавершенным, и судить о его функционале становится труднее.

После подведения данных трех итогов, можно эмпирически их объединить и сделать вывод о том, что все три свойства прототипов сложных инженерных систем – логическая продуманность, интерактивность и графическая завершенность – являются важными для потенциальных заказчиков, тестировщиков и разработчиков [6].

Литература

1. Boehm B.W., Gray T.E. and Seewaldt T. Prototyping versus specifying a multi-project experiment // IEEE Transactions on Software Engineering. – 1984. – V. 10(3). – P. 290–303.
2. Каллауус М.М., Перепелица Ф.А., Погорелов В.И. Сравнительный анализ инструментов прототипирования интерфейса инженерных систем // Альманах научных работ молодых ученых Университета ИТМО. – 2016. – Т. 2. – С. 251–253.
3. Goma H. The impact of rapid prototyping on specifying user requirements // ACM Software Engineering Notes. – 1983. – V. 8(2). – P. 17–28.
4. Bemstein L. Get the design right! // IEEE Software. – 1993. – V. 10. – № 5. – P. 61–63.
5. Ericksson T. Notes on design practice: Stories and prototypes as catalysts for communication. In J. M. Carroll (ed.) // Scenario-based design envisioning work and technology in system development. – 1995. – P. 37–58.
6. Arvola M., Artman H. Enactments in interaction design: How designers make sketches behave // Artifact. – 2007. – V. 1. – № 2. – P. 106–119.



Катышева Полина Андреевна

Год рождения: 1995

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: p.katysheva@gmail.com



Государев Илья Борисович

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

e-mail: ilia.gossoudarev@gmail.com

УДК 004.822

ЭВОЛЮЦИЯ И ПРОБЛЕМЫ ИСПОЛЬЗОВАНИЯ МИКРОРАЗМЕТКИ В ВЕБЕ

Катышева П.А., Государев И.Б.

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

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

В работе рассмотрены особенности применения микроразметки, приведены примеры использования различных словарей и синтаксисов. Также проведен сравнительный анализ

поддержки микроразметки двумя наиболее известными поисковыми системами – Яндекс и Google.

Ключевые слова: микроразметка, словари микроразметки, семантический веб, JSON-LD.

Тенденция Веба первого поколения основывалась на том, что информация, которая содержится нем, будет обрабатываться только человеком. Однако современные технологии стремятся к тому, чтобы все большую часть рутинной работы выполняла вычислительная техника. Это обуславливает то, что Веб нового поколения должен обеспечивать возможность автоматизированной обработки и интерпретации данных. При таких запросах уже недостаточно располагать лишь синтаксисом XML-документов и HTML-страниц, являющихся основой Всемирной паутины, описанных с помощью DTD или XML Schema.

Все это приводит к развитию так называемой семантической паутины или семантического веба, главными компонентами которого являются машиночитаемые элементы – узлы семантической сети, опирающейся на онтологии [1]. Это способствует тому, что программы-клиенты имеют возможность получать информацию вида «предмет – тип связи – другой предмет» и производить на их основе логические заключения.

Семантическая паутина, функционирующая на основе Всемирной, также работает посредством использования протокола передачи гипертекста (HTTP) и единообразного идентификатора ресурсов (URI) [2]. Но, в отличие от Всемирного веба, семантический веб представляется сетью информационных узлов, связанных друг с другом так, чтобы информация была в виде, пригодном для обработки и анализа вычислительной системой.

Одним из средств сообщения роботу поисковой системы о контенте страницы, является добавление семантической разметки, или микроразметки. Микроразметка является дополнительным набором тегов и атрибутов, добавляемых на страницу для описания семантической составляющей информации, заключенной внутри HTML-элементов. Благодаря этим атрибутам, содержимое веб-страниц становится машиночитаемым, что позволяет находить, извлекать и анализировать нужные пользователю данные автоматически.

Словари и синтаксисы. В зависимости от конкретной задачи, которую будет решать внедряемая микроразметка, производится выбор словаря и соответствующего синтаксиса. Словарь представляет собой набор свойств и классов, использование которых позволяет указывать суть содержимого на странице. Синтаксис, в свою очередь, определяет правила использования словаря. Синтаксис описывает, какие теги будут определять ту или иную сущность и ее свойства.

Существует довольно много словарей и синтаксисов, так как семантическая разметка создавалась поэтапно разными группами разработчиков для решения собственных, зачастую узконаправленных, задач [3]. Так, например, словарь OpenGraph [4].

OpenGraph (OG) – словарь, созданный командой разработчиков Facebook для того, чтобы любой информационный веб-ресурс мог стать частью этой сети и имел привлекательный внешний вид внутри нее. С помощью OG размечаются и отображаются расширенные ссылки сайтов.

Данный словарь довольно прост в применении – для его внедрения нужно четыре свойства:

1. og:title – наименование объекта;
2. og:type – тип объекта. В зависимости от типа можно указывать другие свойства;
3. og:image – URL-изображения, описывающего объект;
4. og:url – канонический URL-объекта, используемый в качестве постоянного ID.

Schema.org [5] – разработкой данного словаря занимаются крупнейшие поисковые системы, что позволяет вебмастерам применять на странице разметку одного типа для всех систем, а не свою для каждой. С помощью разметки Schema.org сайты имеют возможность вывести расширенное описание, оформленное специальным образом, в поисковой выдаче.

В данном словаре существует обобщенный тип сущности – Thing, имеющий подтипы. Наиболее распространенные из них:

- Action. Предназначен для описания действия, которое выполняется определенной личностью или организацией. У этого действия может быть указано место, объект и инструменты, с помощью которых оно было осуществлено. Также у него может быть результат, участники и период времени, затраченного на его совершение. С помощью класса Action развиваются Яндекс.Острова, а также реализован проект GmailActions;
- CreativeWork. Служит для описания особенности творческих работ. С применением этого типа могут быть описаны видеоролики, картинки, рецепты и др. Каждая работа может содержать краткое описание, информацию об авторе и жанре, упоминания, отзывы и т.д.;
- Product. Служит для описания товара или услуги. Некоторые платные сервисные услуги (как, например, стрижка) тоже могут быть описаны типом Product. Для товара указываются рейтинг, бренд, цвет, аудитория, цена, модель и т.д.;
- Person. В документации к словарю Schema.org отмечено, что, используя этот подтип, может быть описан любой человек – живущий ныне, умерший или выдуманный. Описание человека может содержать контактные данные, информацию о месте работы и занимаемой должности, семье, отношениях и многое другое.

Микроформаты созданы в 2007 году группой разработчиков, входящих в Консорциум Всемирной паутины (World Wide Web Consortium, W3C), которые намеревались разработать собственный стандарт с использованием базовых элементов языка HTML. Это было определенным преимуществом стандарта, так как его внедрение было проще, в сравнении с другими словарями, но сейчас делать разметку с использованием микроформатов не проще, а иногда и сложнее, чем с другими словарями. В сравнении с OpenGraph и Schema.org, его используют все реже. Microformats.org являются открытым стандартом, что дает возможность любому разработчику принять участие в развитии словаря. Отличие микроформатов от других словарей состоит в том, что они являются объединенным стандартом словаря и синтаксиса.

В настоящее время поисковыми системами поддерживаются следующие микроформаты:

- hCard – предназначен для разметки контактной информации (адресов, телефонов и т.д.);
- hRecipe – служит для описания кулинарных рецептов;
- hReview – является форматом разметки отзывов;
- hProduct – служит для описания товаров.

Наряду со словарями существует несколько стандартов синтаксиса. Наиболее часто используемые:

- Microdata – микроданные. Для внедрения словаря Schema.org наиболее часто используют именно этот синтаксис;
- Microformats.org – являются объединенным стандартом синтаксиса и словаря;
- RDFa и RDFaLite. В упрощенном виде RDFa рекомендуется к использованию разработчиками словаря OpenGraph. Также встречается с другими словарями, например, со словарем DublinCore или DataVocabulary;
- JSON-LD – расширение JSON.

Microdata (микроданные). Синтаксис микроданных предполагает использование атрибутов `itemscope`, `itemtype` и `itemprop`, которые служат для указания на сущность и ее свойства.

Microformats.org (микроформаты). Особенность микроформатов – это применение тегов с атрибутами `class`, `rel`, `rev` или `title` для передачи семантической информации. Для обозначения сущностей и их свойств используются только уже существующие атрибуты и не вводятся новые.

Формат JSON-LD позволяет указать данные в скрипте. Благодаря этому разметку видят только поисковые роботы, для которых она и предназначена.

Поддержка различными поисковыми системами. В конце октября 2016 года специалист по работе с вебмастерами крупнейшего поисковика в мире Джон Мюллер сообщил, что поисковая система Google предпочитает микроразметку, оформленную с помощью синтаксиса JSON-LD. Несмотря на это, поисковая система Google поддерживает и все остальные стандарты синтаксисов. Это значит, что, если на сайте уже имеется микроразметка в формате, отличном от JSON-LD, он также получит расширенный сниппет. Стоит отметить, что в настоящее время, микроразметка не используется поисковиками в качестве сигнала ранжирования.

Поисковая система Яндекс на текущем этапе развития поддерживает формат JSON-LD только в почтовых сообщениях, но пока не отображает данные, представленные с помощью этого формата, в результатах поиска. Это приводит к тому, что может понадобиться размечать одни и те же данные в разном формате синтаксиса.

Выводы. В заключение настоящей работы сделан вывод, что использование микроразметки на сегодняшний день является целесообразным и необходимым действием, направленным на продвижение сайта в системах поиска. Разметка данных с использованием семантических элементов помогает роботам поисковых систем лучше понимать смысл сайтов, а пользователям становится удобнее извлекать из них полезную информацию.

Литература

1. Гладун А.Я., Рогушина Ю.В. Онтологии в корпоративных системах. Часть I // Корпоративные системы. – 2006. – № 1. – С. 41–47.
2. SemanticWeb [Электронный ресурс]. – Режим доступа: <http://www.w3.org/standards/semanticweb/>, своб.
3. Документация микроформатов [Электронный ресурс]. – Режим доступа: <http://microformats.org/>, своб.
4. Документация словаря OpenGraph [Электронный ресурс]. – Режим доступа: <http://ogp.me/>, своб.
5. Документация словаря Schema.org [Электронный ресурс]. – Режим доступа: <http://schema.org/>, своб.



Кирикова Юлия Валерьевна

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4206

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: kirikovayv@gmail.com

УДК 004.51

МЕТОДЫ ОЦЕНКИ ЭРГОНОМИЧЕСКИХ ПОКАЗАТЕЛЕЙ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ВЕБ-ПРИЛОЖЕНИЙ

Кирикова Ю.В.

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

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе рассмотрены понятия эргономики, юзабилити-тестирования, приведена классификация методов тестирования, рассмотрены достоинства и недостатки методов.

Ключевые слова: эргономика, веб-приложения, юзабилити, юзабилити-тестирование, экспертная оценка, удаленное тестирование.

Совершенствование компьютерных технологий, развитие технологических решений для бизнеса, образования, социальных взаимодействий, увеличение времени, проводимое за мобильными устройствами и компьютерами, приводят к росту числа различных веб-приложений и сервисов. Товары и услуги приобретаются через интернет посредством взаимодействия с веб-приложениями. Однако сравнительно небольшой процент таких взаимодействий полностью удовлетворяет пользователей, так как возникают проблемы поиска контента или страницы, совершаются ошибки. Иными словами, часто веб-приложение не удовлетворяет требованиям эргономики, в том числе юзабилити.

Международная ассоциация эргономики (IEA) определяет эргономику как научную дисциплину, изучающую взаимодействие человека и других элементов системы, а также как сферу деятельности по применению теории, принципов, данных и методов этой науки для обеспечения благополучия человека и оптимизации общей производительности системы.

Под веб-приложениями понимаются веб-сайты, в которых пользователь является активным участником при работе с сайтом (например, социальные сети и интернет-магазины).

Согласно стандарту ИСО 9241-11 [1], юзабилити – это степень, с которой продукт может быть использован определенными пользователями при определенном контексте использования для достижения определенных целей с должной эффективностью, продуктивностью и удовлетворенностью [1].

Юзабилити имеет много компонентов и связано с признаками:

- обучаемость;
- эффективность;
- запоминаемость;
- ошибки;
- удовлетворенность.

Для того чтобы оценить эти параметры, проводится юзабилити-тестирование.

Юзабилити-тестирование – это набор методов и инструментов, которые позволяют определить, удобен и понятен ли объект тестирования для предполагаемого применения. Тестирование основывается на привлечении пользователей для оценки уровня реализации пользовательских свойств продукта [2].

В ходе тестирования изучается насколько быстро и эффективно пользователи выполняют конкретные задачи, и с какими проблемами они сталкиваются. Иначе говоря, юзабилити-тестирование – это метод оценки эффективности существующих идей и инструмент для их совершенствования.

Согласно Якобу Нильсену [3], тестирование необходимо производить на всех этапах разработки продукта для более четкого контроля над проектом. Он различает полное тестирование (проводится на последнем этапе разработки продукта) и промежуточное тестирование (проводится в ходе проектирования веб-приложения).

Юзабилити-тестирование необходимо проводить для предварительной проверки на пользователях с целью получения сведений об удовлетворенности продуктом.

В тестировании проверяют проходимость пользовательских сценариев, впечатление от системы, понятность навигации (успешность выполнения заданий на поиск предмета/функции/каталога), полноту и доступность контента, восприятие дизайна.

Существует несколько методик тестирования юзабилити:

- неформальные сеансы обратной связи;
- коридорное тестирование;
- модерируемое удаленное тестирование;
- немодерируемое удаленное тестирование;
- экспертная оценка;
- А/В тестирование.

Неформальные сеансы обратной связи – тестирование, при котором поясняются идеи и выслушиваются соображения пользователя. Как правило, проводятся с коллегами по проекту для быстрого тестирования.

При коридорном тестировании респондент выполняет задания, которые предоставляет модератор, на компьютере, планшете или мобильном устройстве. Часто используется специальное программное оборудование (ПО) и оборудование для записи сессий.

Виды коридорного тестирования:

- тестирование в оборудованной лаборатории (лабораторное тестирование);
- выездные сессии;
- вовлеченное наблюдение за пользователем на его рабочем месте.

Достоинствами такого подхода является: запись всего тестирования, контроль соответствия респондентов целевой аудитории, любые качественные метрики.

К недостаткам относятся: дороговизна проведения, сложность отбора респондентов, эффект Хортон (условия новизны и интереса к эксперименту, повышенное внимание к самому исследованию приводят к искажениям и уходу от реального положения вещей).

Модерируемое удаленное тестирование проводится 1 на 1 (по скайпу, телефону и т.п.), респондент выполняет задания со своего компьютера, подключившись к совместному рабочему столу.

Достоинства: хорошая географическая выборка, запись экрана и голоса, контроль качества выборки респондентов, наличие качественных метрик.

Недостатки: технические сложности, нет наблюдения за респондентом.

Удаленное немодерируемое тестирование осуществляется посредством выполнения тестов, созданных в одной из специальных систем, самостоятельно. Задания формируются для проведения удаленного тестирования. Ссылка на тест предоставляется респондентам, и они самостоятельно проходят тест.

Достоинства: хорошая географическая выборка, статистические данные, сравнительная дешевизна тестирования.

Недостатки: нет контроля соответствия респондента всем критериям, ограничения на сценарий тестирования (короткие задания), нет возможности наблюдения.

Экспертная оценка – один или несколько экспертов проводят аудит системы по заданным сценариям и метрикам, оценивают удобство системы и фиксируют найденные проблемы и дают свои рекомендации.

Достоинства: результат с первого дня, предлагает варианты решения выявленных проблем.

Недостатки: нужен эксперт, субъективная оценка.

А/В тестирование – автоматизированная проверка двух или более версий одного контента на одинаковых аудиториях. Несколько версий контента (страницы, изображения, письма) с незначительными различиями показываются большим группам пользователей. По результатам статистической выборки измеряются показатели каждой версии (конверсия в целевое действие/время, затраченное на просмотр или поиск информации и т.п.).

Достоинства: тестирование проводится в реальных условиях на целевой аудитории, можно тестировать незначительные изменения, дешево.

Недостатки: проводится на готовом веб-приложении с посетителями, нет возможности судить о поведении пользователей на всем сайте.

При многообразии методов юзабилити-тестирования перед руководителем проекта возникает проблема выбора вида тестирования. Ограничениями выступают время, бюджет, а также особенности целевой аудитории. Решением проблемы является сочетание видов юзабилити-тестирования. Например, на первом этапе проводятся неформальные сеансы обратной связи, далее веб-приложение оценивается экспертом (на этом этапе исправляется большинство проблем), после проводится тестирование на пользователях: коридорное, модерлируемое удаленное или немодерлируемое удаленное тестирование. При внесении изменений в готовое веб-приложение, проводится А/В тестирование.

Литература

1. ГОСТ Р ИСО 9241-210-2012. Эргономика взаимодействия человек-система. Часть 210. Человеко-ориентированное проектирование интерактивных систем. – Введен 01.12.2013. – М.: Стандартиформ, 2013. – 32 с.
2. Сергеев С.Ф. Методы тестирования и оптимизации интерфейсов информационных систем: учебное пособие. – СПб.: НИУ ИТМО, 2013. – 117 с.
3. Нильсен Я., Лоранжер Х. Web-дизайн: удобство использования Web-сайтов. – М.: Вильямс, 2007. – 368 с.



Костюченко Юлия Андреевна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: jukostyuchenko@gmail.com



Государев Илья Борисович

Год рождения: 1979

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна,

к.педагог.н., доцент

e-mail: ilia.gossoudarev@gmail.com

УДК 004

ПРОБЛЕМЫ ВЕБ-СЕМАНТИКИ В ОБЛАСТИ ПРОДВИЖЕНИЯ ВЕБ-РЕСУРСОВ

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 рассмотрена семантическая разметка веб-страниц, выделены проблемы, связанные с продвижением веб-ресурсов, обозначаются HTML-элементы, облегчающие поисковую оптимизацию сайтов.

Ключевые слова: веб-семантика, разметка, SEO, продвижение, HTML, микроданные, мета-теги.

В последнее время особенно актуальной стала проблема поиска релевантной информации в интернете. Объем информации увеличивается с каждым днем, поэтому необходим поиск, основанный не только на содержании страниц, но и на семантическом значении информации.

Неотъемлемой частью современного интернета является SEO (Search Engine Optimization) – оптимизация сайтов для поисковых систем. Комплекс мер для повышения позиций сайта в результатах выдачи поисковых систем позволяет существенно повысить посещаемость ресурса и финансовый доход от него. Для более грамотного продвижения используется семантическое ядро, которое позволяет качественнее описать контент с помощью ключевых слов и правильно оптимизировать сайт для индексации поисковыми системами.

Поисковые машины облегчают сбор информации путем предоставления перечня ресурсов, ранжированного согласно субъективной релевантности (т.е. «с точки зрения поисковой машины»), который можно также отнести к своему роду интегрированному представлению информации об объекте запроса. Для того чтобы сделать поиск информации в интернете более удобным и структурированным, необходимо использовать в разметке сайта семантические инструменты [1].

Семантика сайта – это организация структуры сайта, в которой содержится набор ключевых слов, отвечающих запросу пользователя. Особенность состоит в том, что все ключевые слова образуют связанную структуру. Они объединены и связаны для обеспечения большего охвата всевозможных запросов пользователя – это и есть семантическое ядро сайта. В нем отражаются все запросы, обрабатываемые сайтом, закрывая все «дыры» в ядре, которые могут возникать при недостаточном охвате ключевых слов.

При написании HTML-кода задается структура контента. На данном этапе определяются абзацы, списки, таблицы и заголовки. Написание семантического кода подразумевает подбор правильного элемента для определения той или иной структуры. У каждого элемента есть свои общепризнанные значения и функции.

HTML также задает смысл контента. У большинства HTML-элементов есть неявная роль, определяющая назначение элемента. Например, неявная роль тега `<a>` – «ссылка», у `` – «графика» или «изображение», а у `<header>` – «баннер». Разные API-доступности, поддерживаемые платформой, могут распознавать эти роли, а вспомогательные технологии вроде инструментов распознавания речи и скринридеров пользуются ими.

Семантическую разметку также называют микроразметкой – это разметка страницы с дополнительными тегам и атрибутами в тегах, которые указывают поисковым роботам на то, о чем написано на странице. Она дает несколько преимуществ [2]:

- код лучше читается и воспринимается поисковыми машинами: страница, которая имеет семантическую разметку, выдается в поисковых системах выше, чем страница с несемантическим кодом;
- код более понятен для человека: код, в котором все четко прописано, легче редактируется и воспринимается;
- большая гибкость элементов: с помощью специальных средств с семантическим кодом гораздо проще обращаться к определенным элементам, изменять их;

– простота поиска: пользователю не нужно формулировать запрос, исходя из того, как его воспринимает компьютер; он может сформулировать его так, как ему удобно.

Когда поисковые системы индексируют тот или иной сайт, они интерпретируют содержание страниц сайта на основе разметки. Полученные данные могут быть использованы для лучшей индексации контента (соответственно, улучшения его продвижения), а также при голосовых ответах и в картах. Языковые инструменты (переводчики) исследуют разметку для перевода на другой язык. Грамотная HTML-разметка может привести к более точному переводу. Социальные веб-сервисы (Facebook, Twitter, Pinterest) берут части статей с различных сайтов и отображают их на своей платформе. Если правильно использовать семантику, то выдача будет более точной, что приведет к большему количеству пользователей на продвигаемый ресурс (как следствие, увеличит прибыль) [3].

В данной работе для примера семантической верстки использовалась структурированная разметка данных Schema.org. Это совместный проект Google и Yahoo для предоставления способа лучшего понимания содержимого веб-страниц поисковыми системами. Данная разработка дает сайтам возможность получать специальные сниппеты в поисковой выдаче.

Для осмысленного структурирования используются следующие HTML-элементы (по спецификации W3C) [4]:

- `article` – элемент представляет собой обособленный фрагмент контента, который может существовать сам по себе;
- `header` – вводный контент лучше структурировать как дочерний элемент данного. Это может быть заголовок, дата публикации и (или) имя автора;
- `footer` – так называемый «подвал»; обычно содержит сведения о разделе, ссылки на различные документы, авторские данные и прочее.

Для усиления семантической HTML-структуры применяются микроданные. Микроформаты – это один из словарей микроразметки, как и Schema.org, Open Graph или FOAF. Единственное отличие в том, что микроформаты представляют собой объединенный стандарт синтаксиса и словаря, тогда как микроразметка – собирательный термин для способа обогащения страницы семантическими данными [5].

Краткое описание некоторых микроданных:

- `name` – указывает на имя пункта; свойство `name` – это заголовок веб-страницы, который представлен в элементе `title`;
- `headline` – человекочитаемое название статьи. Некоторые сайты используют короткие, богатые ключевыми словами значения для `title` из-за SEO, а затем идет полный заголовок, описывающий тему статьи;
- `description` – краткое объяснение описания статьи; присвоенное данному тегу значение обычно хорошо работает;
- `author` – имя создателя контента;
- `about` – это свойство применимо к тексту, описывающему тему статьи. Подходит для сопровождающего предложения или абзаца.

Также широко используется оптимизация при помощи мета-тегов. Изначально они предназначались для того, чтобы предоставлять поисковым системам информацию о контенте сайта. Остановимся на основных – `Robots` и `Description`.

Мета-тег `Robots` используется для управления работой поисковых роботов на уровне страницы. Ему можно присвоить одно из следующих значений [6]:

- `index/noindex` – разрешает/запрещает поисковым системам индексировать страницу. По умолчанию индексируются все страницы, поэтому «`index`» можно не указывать;
- `follow/nofollow` – разрешает/запрещает поисковым системам переходить по ссылкам со страницы. По умолчанию всем страницам присваивается значение «`follow`»;

- noarchive – запрещает поисковым системам сохранять кэшированную копию страницы. По умолчанию сохраняются копии всех проиндексированных страниц;
- nosnippet – запрещает поисковым системам отображать блок с описанием (сниппет) в результатах поиска.

Мета-тег Description представляет собой краткое описание содержания страницы. Поисковые системы не учитывают ключевые слова или фразы из этого тега при ранжировании сайтов, но эта информация чаще всего используется в качестве текста для сниппета в результатах поиска. Создание читабельного и привлекательного описания с использованием ключевых слов может существенно увеличить кликабельность ссылки. Описание может быть любой длины, но следует учесть, что поисковые системы отображают только первые 160 символов, а остальной текст обрезают.

От того, как составлены title и Description, зависит и ранжирование сайта поисковыми системами, и его посещаемость. Если с точки зрения поисковых роботов title будет максимально релевантным (соответствующим по смыслу) запросу, сайт при прочих равных получит более высокую позицию, чем веб-проекты конкурентов.

Семантическая разметка – перспективное направление, которое не лишено проблем на данном этапе развития. Только серьезный подход к разметке, оптимизация мета-тегов и использование различных семантических инструментов, а также внимание к интересам потенциальных посетителей вашего веб-ресурса, позволят эффективно продвигать тот или иной сайт.

Литература

1. Витько А.В., Стрелец Л.Р. Методология внедрения семантики в веб-сайт // Восточно-Европейский журнал передовых технологий. – 2013. – № 2(57). – С. 31–33.
2. Рябов В.А., Несвижский А.И. Семантические технологии и микроформаты [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/department/internet/mwebtech/20>, своб.
3. Геушева И.С., Фаустова К.И. Современные методы продвижения сайтов в интернете // Территория науки. – 2016. – № 4. – С. 121–124.
4. World Wide Web Consortium [Электронный ресурс]. – Режим доступа: <https://www.w3.org>, своб.
5. Фаустова К.И. Значение SEO для эффективных продаж в интернете // Территория науки. – 2015. – № 3. – С. 139–144.
6. Жданов К.Е. Семантика в HTML5 [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/49734>, своб.



Красноруцкая Надежда Сергеевна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4206

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: Nadegda.Sergeevna@mail.ru



Перепелица Филипп Александрович

Год рождения: 1977

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, ст. преподаватель
e-mail: perepelitcafa@niuitmo.ru



Шалобаев Евгений Васильевич

Год рождения: 1947

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, к.т.н., доцент
e-mail: kpd@limtu.ru

УДК 681.618.9/621.01

АДДИТИВНЫЕ ТЕХНОЛОГИИ ДЛЯ ИЗГОТОВЛЕНИЯ СЛОЖНЫХ ДЕТАЛЕЙ В МАШИНОСТРОЕНИИ

**Красноруцкая Н.С., Перепелица Ф.А., Шалобаев Е.В.
Научный руководитель – к.т.н., доцент Шалобаев Е.В.**

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе перечислены преимущества аддитивных технологий, внедренные в производство компаний в различных отраслях машиностроения. Подробно рассмотрены этапы компьютерного проектирования нетрадиционной продукции в редуكتورостроении с последующим ее изготовлением при помощи аддитивных технологий.

Ключевые слова: аддитивные технологии, AF/AM-технологии, машиностроение, редуكتورостроение, САД-система, 3D-модель, прототип изделия.

В начале 1980-х гг. начали развиваться новые методы производства деталей, основанные на послойном выращивании изделия по модели систем автоматизированного проектирования, называемые ныне аддитивными технологиями (AF- или AM-технологии) [1, 2].

Аддитивные технологии отличаются друг от друга выбором материалов и способом нанесения, однако во всех случаях создание модели основывается на поочередном наращивании слоев. Данные технологии используются в авиационной, автомобильной промышленности, космической индустрии и др. областях машиностроения.

Наибольший интерес к непосредственному «выращиванию» металлических изделий (технология DMF) проявляют в аэрокосмической промышленности и оборонных отраслях. По заявлению К. Фурстосса, руководителя группы производства и технологии материалов компании General Electric, через 10 лет примерно половина деталей энергетических турбин и авиационных двигателей будет изготавливаться с помощью AM-технологий. Аналогичную позицию занимают и российские ученые из ВИАМ.

Использование аддитивных технологий позволяет значительно увеличить производительность изделий. Так, компания General Electric успешно использует промышленные 3D-принтеры для производства деталей реактивных двигателей. Согласно заявлениям представителей исследовательского центра компании, топливные форсунки реактивных двигателей, изготовленные таким образом, на 25% легче и в пять раз прочнее сопла, изготовленного традиционным способом – при помощи сборки более чем двадцати различных элементов. Очевидной является и экономическая эффективность этих инноваций: сокращение производственных расходов на предприятиях компании благодаря аддитивным технологиям составляет около 30%. Представители General Electric заявляют, что применение 3D-печати позволит на 15% уменьшить потребление топлива самолетами, за год экономия составит для компаний 1 млн долл. США на одном самолете.

Еще одним преимуществом аддитивных технологий является сокращение времени разработки изделий из-за высокой скорости изготовления. Руководство Ford Motor заявляет, что применение 3D-принтеров для печати прототипов тормозных дисков и головок цилиндров обеспечивает экономию рабочего времени на производстве до 40%.

Сложная конфигурация, которую нельзя получить механической обработкой, может быть легко воспроизведена селективным нанесением материала [3].

Процесс проектирования нетрадиционной продукции в редукторостроении с последующим ее изготовлением при помощи аддитивных технологий имеет ряд особенностей. Эти особенности хорошо видны в сравнении с проектированием изделия для последующего литья.

Производство изделия посредством аддитивных технологий не требует создавать специальную технологическую оснастку, как дорогостоящие литьевые формы, и тем самым удешевляют производство.

Впрочем, и сами литьевые формы могут быть выращены вместе с каналами охлаждения произвольной формы, что невозможно сделать при обычных методах механообработки [1, 2].

Алгоритм процесса проектирования нетрадиционной продукции в редукторостроении с последующим ее изготовлением при помощи аддитивных технологий наглядно представлен на рисунке (переработка алгоритма в работе [4]).

Сущность AF-технологий состоит в следующем: создание 3D-модели стандартного изделия в САД-системе → 3D-модели нестандартного изделия в САД-системе → Создание прототипа → Выбор материала → Анализ изделия с учетом материала → AF-машина → Готовое изделие (модифицированное относительно работ [4, 5]).

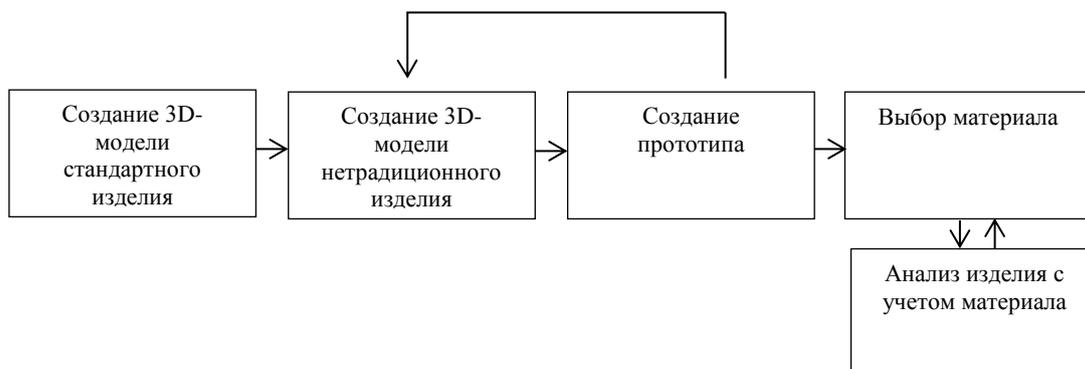


Рисунок. Этапы компьютерного проектирования нетрадиционной продукции в редукторостроении

– Первый этап. Использование 3D-модели стандартного изделия. Стандартная модель создается в базовых САД-системах с использованием стандартного функционала, таких как Autodesk Inventor, Autodesk Fusion 360 или в специализированных модулях.

В качестве инструмента создания 3D-модели можно использовать систему трехмерного моделирования КОМПАС-3D от Российской компании АСКОН. В отличие от многих подобных отечественных разработок АСКОН создал собственное математическое и геометрическое ядро, что позволяет производить тонкую настройку программ, в том числе и в сфере безопасности.

- Второй этап. Создание 3D-модели нетрадиционного изделия. Создание, а точнее адаптация, 3D-модели стандартного изделия с учетом требований предъявляемых нетрадиционному изделию.
- Третий этап. Создание прототипа. Полученная 3D-модель визуализируется с помощью обычного 3D-принтера при этом создается прототип изделия [4].

Прототип изделия можно использовать в качестве концептуальной модели для визуализации и анализа конструкции. Полученный прототип позволит выполнить доработку конструкции, провести функциональные тесты, а также может служить мастер-моделью для изготовления технологической оснастки.

При быстром прототипировании, которое предлагают большинство недорогих AF-принтеров, следует учитывать, что качество прототипа оставляет желать лучшего.

После создания прототипа и его анализа при необходимости можно вернуться к второму этапу для уточнения параметров нетрадиционного изделия.

- Четвертый этап. Выбор материала. Выбор материала является ключевой задачей при проектировании новых изделий, а применение систем виртуального моделирования помогает не просто выбрать материал, но и спроектировать требуемый, например, полимерный композиционный материал.

В принятии решений участвуют различные специалисты (представители заказчика, дизайнеры, конструкторы и технологи, материаловеды, метрологи), образующие экспертную группу, которая отвечает за принятие окончательного решения.

Подбор материала осуществляется на основе международных открытых баз данных материалов, таких как CAMPUS, MatWeb, IDES и т.п., а также аналогичных баз специализированных научных организаций.

При подборе материала можно использовать базу «Справочник Материалы и Сортаменты» отечественного разработчика программного обеспечения компании АСКОН.

- Пятый этап. Анализ изделия с учетом материала. Применение к адаптированной 3D-модели изделия из выбранного материала позволяет провести оптимизацию и проверку проектов с помощью специализированного программного обеспечения. Например, программный комплекс Autodesk Simulation позволяет проводить структурный анализ, предоставляет информацию о номинальной толщине стенок, величине уклона, стоимости материалов и т.д.

После проведения анализа 3D-модели изделия с учетом конкретного материала при отрицательном результате необходимости можно вернуться к этапу выбора материала.

Применение металлозамещающих пластмасс в аддитивных технологиях, например, для производства зубчатых колес в сочетании оригинальных кинематических схем многопоточных планетарных редукторов может дать качественный рост надежности трансмиссий в целом [3].

Литература

1. Зленко М.А., Попович А.А., Мутылина И.Н. Аддитивные технологии в машиностроении. СПб.: Изд-во СбГПУ, 2013. – 221 с.
2. Валетов В.А. Аддитивные технологии (состояние и перспективы). Учебное пособие. – СПб.: Университет ИТМО, 2015. – 63 с.

3. Суриков Д.Г. Разработка методики предупреждения отказов механических трансмиссий мехатронных приводов трубопроводной арматуры // *Фундаментальные и прикладные проблемы техники и технологий*. – 2015. – № 5. – С. 121–129.
4. Медунецкий В.М., Романов Н.А. Компьютерное моделирование и визуализация технологии изготовления нестандартных зубчатых колес из полимерных композиционных материалов // *Изв. вузов. Приборостроение*. – 2015. – Т. 58. – № 5. – С. 397–400.
5. Зленко М.А., Нагайцев М.В., Довбыш В.М. Аддитивные технологии в машиностроении. Пособие для инженеров. – М.: Изд-во «НАМИ», 2015. – 220 с.



Кузнецова Полина Борисовна

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4206

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: findkosh@gmail.com

УДК 004.921

МЕТОДИКА ПОСТРОЕНИЯ 3D-МОДЕЛИ ОБЪЕКТА НА ОСНОВЕ ДАННЫХ СВЕТОВЫХ ПОЛЕЙ

Кузнецова П.Б.

Научный руководитель – д.т.н., профессор Погорелов В.И.

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе затронуты вопросы методики построения 3D-модели объекта на основе данных световых полей пленоптической камеры, описаны основные подзадачи, оптимальные решения которых следует использовать или доработать существующие, а также предложены решения для первых этапов, изложен план дальнейшей работы.

Ключевые слова: 3D-модель, карта глубины, пленоптическая камера, световое поле, дефокусировка, диспаритет.

С момента появления фотографии прошло уже почти 200 лет, однако, до сих пор современная фотография имеет определенные проблемы. Одна из главных заключается в том, что фотография – двухмерное изображение, тогда как сейчас хочется иметь информацию об изображенных объектах, в частности, об их удаленности от фокальной плоскости, о форме и относительных размерах объекта. Действительно, если иметь информацию о глубине, тогда из обычной фотографии можно создавать 3D-модели объектов, чтобы затем использовать их, например, в приложениях дополненной реальности. Пленоптические камеры, появившиеся относительно недавно, позволяют фиксировать больше информации о распределении поступающего света и получать фотографии, сфокусированные на разных объектах. Одной из таких камер является Lytro [1].

Однако у этой камеры есть недостаток. Разрешение устройства первого поколения составляет всего лишь 20×20 пикселей, поэтому для эффективного использования при моделировании необходимо применять алгоритмы улучшения изображения и построения карты глубины.

При решении основной задачи возникают две глобальные подзадачи: анализ методов получения карты глубины и непосредственно нахождение оптимального способа описания поверхности объекта.

На сегодняшний день среди основных алгоритмов вычисления карты глубины заслуживающих внимания: метод дефокусировки и метод вычисления диспаратета. В данной работе рассмотрены эти алгоритмы.

Метод дефокусировки. Как говорилось ранее, Lytro позволяет получать из одной фотографии набор изображений, сфокусированных на разной глубине. Таким образом, можно получить информацию о глубине точек на основе исследования их четкости относительно ряда изображений.

Прежде всего, набор изображений должен быть получен из так называемого raw («сырого») файла. У каждого изображения в наборе есть свое значение глубины. Так как программное обеспечение камеры производит весьма грубую карту глубины размером 20×20 пикселей, требуются более точные вычисления.

Для решения этой проблемы необходимо оценить, насколько хорошо произведена фокусировка в пикселях каждого из изображений. Чем больше контраст и меньше размытие, тем лучше фокусировка в данной области. Сравнивая значения контраста всех точек среди изображений в наборе, определяем, в каком из них они лучше всего сфокусированы. Найдя такое изображение, приравниваем значение его глубины к соответствующему пикселю и в сумме получаем карту глубины [2].

Для определения значения контраста используется оператор Лапласа. Этот оператор часто используется в обработке изображений для выделения границ.

После его применения к каждому изображению в наборе, карта глубины может быть сгенерирована по основе значения глубины каждого пикселя из того изображения, где в нем сравнительно лучшая фокусировка.

Метод диспаратета. Фотографию светового поля можно представить в виде набора изображений с различными точками обзора за счет перегруппировки пикселей исходной фотографии. Такие изображения называются субапертурными.

Этот факт позволяет применить алгоритмы сопоставления стереоизображений: вычисления карты диспаратета, из которой, благодаря линейным преобразованиям, может быть получена карта глубины [3]. Диспаратет (соответствие) – это расстояние между двумя одинаковыми точками в двух бинокулярных изображениях. Зависимость глубины от диспаратета задается следующим образом [4]:

$$D(x, y) = \frac{f \cdot B}{d(x, y)},$$

где $D(x, y)$ – глубина в точке; f – фокусное расстояние; B – основание стереоизображений (расстояние между ними); $d(x, y)$ – диспаратет в точке.

Общий порядок вычислений:

1. предварительная обработка изображений;
2. вычисление диспаратета в каждой точке;
3. обработка результата фильтром (используется медианный фильтр).

На основе базы изображений Lytro [5] была проведена оценка описанных методов. В качестве метрики использовались медианное абсолютное отклонение получаемых данных глубины от эталонных (условно) и процент ошибок относительно заданного уровня [6]. При использовании метода дефокусировки результат точнее, но с более высокой степенью размытости. Использование метода диспаратета привело к более четкой карте, но при этом значительно менее качественной по значениям глубины.

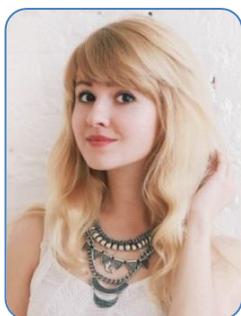
Основными проблемами применения методов являются повторяющиеся однородные и затемненные области – здесь результаты можно считать неудовлетворительными.

Следующим этапом исследования следует обозначить анализ релевантности модели построения поверхности объекта полученным картам глубины и ее выбор.

Кроме того, в дальнейшее исследование должны войти проблемы оптимизации вычислений, так как рассматриваемые этапы являются достаточно ресурсоемкими задачами.

Литература

1. Официальный сайт Lytro [Электронный ресурс]. – Режим доступа: <https://www.lytro.com/>, своб.
2. Lu W., Mok W.K., Neiman J. 3D and Image Stitching With the Lytro Light-Field Camera [Электронный ресурс]. – Режим доступа: https://www.google.ru/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwiFoND35aHVAhXJa5oKHVUZCIMQFggnMAA&url=http%3A%2F%2Fjeremyneiman.com%2Fprojects%2Flytro_report.pdf&usq=AFQjCNFG4A1jK0MVLi9ckDSZIsC5Jiftuw&ad=rjt, своб.
3. Котюжанский Л.А. Вычисление карты глубины стереоизображения на графическом процессоре в реальном времени // Фундаментальные исследования. – 2012. – № 6-2. – С. 444–449.
4. Стерео-реконструкция / Лаборатория компьютерной графики при ВМК МГУ [Электронный ресурс]. – Режим доступа: http://courses.graphicon.ru/files/courses/vision/2010/cv_2010_12.pdf, своб.
5. База изображений Lytro [Электронный ресурс]. – Режим доступа: <http://optics.miloush.net/lytro/TheResources.aspx>, своб.
6. Schechner Y. and Kiryati N. Depth from defocus vs. stereo: how different really are they? // Proc. International Conference on Pattern Recognition. – 1998. – P. 1784–1786.



Куликова Кристина Юрьевна

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4206

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: christina_kulikova@mail.ru

УДК 004.51

ОНЛАЙНОВАЯ ИНФОГРАФИКА: ТИПЫ, ЦЕЛИ, ОСОБЕННОСТИ ДИЗАЙНА

Куликова К.Ю.

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

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе рассмотрено понятие инфографики, проведено сравнение различных видов инфографики и особенностей создания для каждого вида.

Ключевые слова: инфографика, онлайн-инфографика, графический дизайн, визуализация, дизайн информации.

За последние десять лет Интернет стал основным источником информации и новостей. Люди ежедневно сталкиваются с огромным количеством информации,

поэтому возникает проблема: как избавиться от фактического мусора, отфильтровать ценную информацию и запомнить нужный материал. Справиться с этим помогает онлайн-инфографика, являющаяся ответвлением от классической инфографики [1–3].

Изначально инфографика рассматривалась как одно из ответвлений дизайна информации, и значение этого термина сводилось исключительно к визуализации данных. Теперь под термином «инфографика» понимают графический дизайн в широком смысле, одновременно включающий в себя визуализацию данных, подготовку текста и изображений (рисунок). Инфографика напрямую связана с феноменом, называемым «эффект превосходства изображений». Смысл этого эффекта в том, что люди запоминают картинки лучше, чем слова.

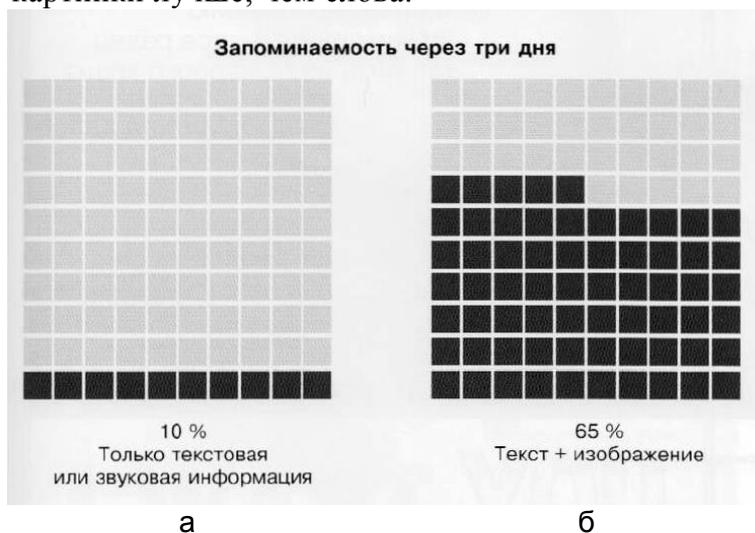


Рисунок. Визуализация данных (а); текст и изображение (б)

Онлайн-инфографика нацелена не столько на эффект запоминания фирмы или человека, ее презентующего, сколько на маркетинговые и PR-цели. Визуализация помогает наращивать трафик, повышает заинтересованность читателя. Процесс создания онлайн-инфографики – это не только использование броских графических элементов. Это прежде всего закономерный процесс, имеющий общие правила, логику, структуру. Создание инфографики объединяет такие области как: графический дизайн, типографика, маркетинг, копирайтинг.

Форматы инфографики. По формату онлайн-инфографика делится на два вида: вертикальная и горизонтальная. Инфографика вертикального формата является инфографикой нового типа. Она приобрела популярность за счет того, что в наше время большинство людей пользуется Интернетом на телефонах. Зритель постепенно усваивает предложенную информацию, которая выстроена по принципу «сверху вниз». Инфографика вертикального формата имеет последовательную компоновку и обладает рядом преимуществ по сравнению с горизонтальной инфографикой. Большинство сайтов скомпоновано так, что информация располагается вертикально и не требует прокрутки по горизонтали. При создании онлайн-инфографики действует общее правило: полномасштабная версия, предназначенная для публикации онлайн – это изображение шириной 800 пикселей. Высота изображения не ограничена. Такой размер позволяет отобразить на сайте оригинальную версию инфографики с оптимальным разрешением, и инфографика будет полностью помещаться по ширине в окне большинства браузеров при любых разрешениях экрана.

Но когда инфографика используется на других сайтах, она обычно уменьшается до миниатюры, чтобы уместиться в ту часть макета, которая отводится под основной контент. (Популярным примером может стать сайт Pinterest). Информация в вертикальном формате

остаётся удобочитаемой, в отличие от горизонтального формата, где при сужении картинки воспринимать информацию становится почти невозможно.

Хоть инфографика в вертикальном формате идеально подходит для Интернета, для других носителей этот вид инфографики не очень удобен. Инфографику в вертикальном формате сложно использовать в презентациях и при распечатке раздаточного материала.

Типы онлайн-инфографики

1. **Справочная инфографика.** Справочная инфографика, как следует из названия, несет в себе информацию о той или иной области, факте, событии. Большинство инфографики, размещаемой онлайн, является справочной. Это связано с тем, что пользователю Интернета нравится смотреть и распространять интересную и познавательную информацию, нежели рекламу.

Примером такой инфографики может стать рецепт блюда, советы путешественнику, гид по городу, биография знаменитости. Эту инфографику с наибольшей вероятностью будут распространять в социальных сетях, размещать на тематических онлайн-площадках.

Справочная инфографика активно используется в блогах. Многие блогеры создают инфографику в едином стиле, что повышает узнаваемость их блога, а внедрение такой инфографики на популярные ресурсы повышает его посещаемость. Ярким примером может стать сайт Pinterest, где по запросу «Нью-Йорк» первым делом выходят небольшие картинки с инфографикой с советами, куда пойти, что посмотреть и что привезти. Такая инфографика сразу обращает на себя внимание. К ней обязательно прикрепляется ссылка на источник – сайт блогера, где находится статья с более подробным описанием города. Таким образом, справочная инфографика отчасти связана с рекламной инфографикой.

2. **Рекламная инфографика.** Рекламная инфографика создается для определенной рекламной компании, главной целью которой является рассказать о товаре, услуге, стимулировать аудиторию на совершение действия. Читатели делятся рекламной инфографикой реже, чем справочной, потому что это может быть воспринято, как навязывание товара, а не сообщение полезной информации. Тем не менее, визуализация данных служит для компании отличным способом демонстрации достоинств товара потенциальным клиентам. Для такой инфографики важно качество для зрителей, а не их количество, ведь главное в такой рекламе – рассказать о товаре, а не увеличить количество ссылок на сайт. На рекламной инфографике можно разместить:

- сравнение товара с аналогами;
- сведения о достоинствах товара;
- спецификацию изделия.

Эффект превосходства изображения позволяет повысить запоминаемость информации на 65%. Интересно составленная инфографика может привлечь внимание клиента на товар или услугу и в последствие сыграть решающую роль при выборе клиента.

3. **Персуазивная инфографика.** Персуазивная инфографика адресует читателю четкий призыв к действию, убеждает совершить действие после просмотра инфографики. Зачастую человек воспринимает любую информацию, подаваемой в «официальной» форме, как достоверную. Действия, к которым визуализация данных подталкивает аудиторию, могут быть разнообразными, например: «Займитесь спортом», «Помогите защитить природу», «Посетите мероприятие» и т.д. Структура такой инфографики построена в соответствии с мотивационной последовательностью Монро. Это классическая методика выстраивания убедительных речей. Пример структуры инфографики, построенной по такой методике:

- ключевое сообщение – содержит информацию, привлекающую внимание зрителя;
- в чем заключается проблема? – содержит обоснование ключевого сообщения;
- опасность – содержит информацию о рисках, читатель узнает, почему эта информация полезна именно для него;
- решение – содержит информацию о предлагаемом решении проблемы;

– что я могу сделать? – содержит информацию о призывах к действию.

4. Визуальное объяснение. Целью такой инфографики является не рассказать о товаре, услуге, а просто визуализировать информацию, идею, процесс, которые сложно объяснить словами. Такая инфографика очень популярна, пользователи склонны сохранять ее и размещаться в социальных сетях, чтобы в дальнейшем использовать в работе. Например, учебные материалы, помогающие лучше запомнить информацию.

Особенности дизайна инфографики. Цель дизайна инфографики отличается от стандартных целей дизайна. Здесь главную роль играет не украшательство, а максимально простое и ясное сообщение информации адресату. Хорошая инфографика эффективно выстраивает связи, стимулирует перепосты в социальных сетях и уникальным образом привлекает внимание к теме.

1. Точность. Самый важный аспект при разработке любой инфографики, все остальное вторично. Визуализация данных должна совпадать с цифрами. Графические элементы могут быть нарисованы приблизительно к данным, но ни в коем случае не должны противоречить им. Например, 30% изображены как 60% и т.д. От точности визуализации данных зависит доверие зрителя, поэтому важно передать все данные верно.
2. Выбор интересной темы. Чтобы заинтересовать читателя, информация, представляемая на инфографике должна быть актуальной или интересной. Трендовые темы – темы, актуальные в определенный период. Если во время уловить тренд, то можно легко привлечь внимание к своей инфографике.
3. Правильное ключевое послание. При создании ключевого послания важно помнить о правиле 5 с. Именно столько времени должно понадобится читателю, чтобы прочитать заголовок и увлечься темой. Большинство читателей просматривают инфографику «по диагонали», поэтому броский заголовок является важнейшей частью дизайна инфографики.
4. Максимальная наглядность информации. Повысить наглядность информации помогает визуализация: диаграммы, графики, иллюстрации, иконки.

Визуализация помогает:

- привлечь внимание зрителя;
- уменьшить количество времени, необходимое зрителю для восприятия данных;
- обеспечить повышенную запоминаемость ключевого послания;
- сделать информацию более доступной для зрителей, говорящих на других языках.

1. Работа с текстом. Хотя инфографика зачастую содержит небольшое количество текста, он должен быть правильно оформлен, чтобы привлекать внимание, побуждать к прочтению. Для этого не нужно использовать слишком крупный шрифт, это не упрощает восприятие информации. Информация, выделенная крупным шрифтом, отвлекает на себя внимание, таким образом, общее содержимое инфографики будет восприниматься разрозненно, зритель будет его осмысливать, отталкиваясь от той информации, которой уже располагает.
2. Логотип компании. Использование логотипа компании не только повышает узнаваемость компании, но и вызывает больше доверия у читателя. Также можно указать ссылку на сайт.
3. Общая стилистика. Используя различные графические элементы для визуализации данных важно выдерживать общий ритм композиции, следить, чтобы инфографика не выглядела перегруженной, аляповатой, негармоничной. Цвета не должны конфликтовать друг с другом, слишком яркие сочетания только отталкивают читателя. Фон и элемент, расположенные на нем должны быть контрастными, не смешиваться и не мешать восприятию информации.

Таким образом, онлайн-инфографика является самым актуальным видом инфографики в настоящее время. Влияние инфографики наблюдается и в веб-дизайне. Столь популярные landing page несут в себе все признаки инфографики: обилие графических элементов, четко структурированная информация, небольшой объем текста, наличие сюжета

(начало, основная часть, заключение), броский заголовок, привлекающая информация. В дальнейшем онлайн-инфографика будет претерпевать и другие изменения, внедряясь в социальные сети, учебные ресурсы, блоги. Онлайн-инфографика будет использоваться и в интерфейсах, заменяя большие объемы текста и облегчая информационную загрязненность.

Литература

1. Meirelles I. Design for information. – Beverly, MA: Rockport Publishers, 2013. – 225 p.
2. Крам Р. Инфографика. Визуальное представление данных. – СПб.: Питер, 2015. – 384 с.
3. Смикиклас М. Инфографика. Коммуникация и влияние при помощи изображений. – СПб.: Питер, 2014. – 152 с.



Лисакова Анастасия Владимировна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: coctail_rossini@mail.ru



Государев Илья Борисович

Год рождения: 1979

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

e-mail: ilia.gossoudarev@gmail.com

УДК 004

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ВОЗМОЖНОСТЕЙ ИСПОЛЬЗОВАНИЯ КОНТЕКСТНОЙ РЕКЛАМЫ ПОИСКОВЫХ СИСТЕМ ЯНДЕКС.ДИРЕКТ И GOOGLE ADWORDS

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 проведено сравнение двух наиболее популярных сервисов для контекстной рекламы. Выявлены их основные характеристики и представлен сравнительный анализ сервисов.

Ключевые слова: контекстная реклама, рекламная кампания, статистика, Google Adwords, Яндекс.Директ.

Контекстная реклама – это вид интернет-рекламы, которая показывается в зависимости от поискового запроса пользователя к поисковой системе (например, в Яндекс или Google), или в зависимости от смысла конкретной веб-странички [1].

Термин «контекстная реклама» появился в 2001 году благодаря сервису Яндекс.Директ. Рекламодателям была предоставлена возможность самостоятельно, без помощи рекламных агентств, размещать текстовые объявления на страницах результатов поиска [1].

Если рассматривать данные за 1 квартал 2016 года, то по данным Ассоциации коммуникационных агентств России, прирост рекламы в интернет-сегменте составил 31% по сравнению с 2015 годом. На данный момент рекламы в интернете являются одними из самых эффективных и популярных способов продвижения бизнеса.

Контекстная реклама базируется на поисковой системе. Основными поисковыми системами в России являются Яндекс и Google. Поисковой системой №1 в Рунете на данный момент является Яндекс. По данным, взятым из открытых источников системы статистики интернет-ресурсов LiveInternet.ru, доля Яндекса на рынке Рунета – 48,3%, Google – 45% [2].

Сравнительная характеристика сервисов контекстной рекламы поисковых систем Яндекс и Google позволит выявить предоставляемые возможности и сделать основные выводы по ним.

Яндекс.Директ – это один из способов размещения поисковой и тематической контекстной рекламы [3].

Работая с Директом, пользователь может:

- самостоятельно определять максимальную цену перехода по объявлениям;
- гибко настраивать собственную рекламу – например, есть возможность задать необходимое время и регионы показа;
- присутствует возможность изучить статистику об эффективности рекламной кампании – по каждому объявлению или по каждой ключевой фразе.

AdWords – сервис контекстной в основном поисковой рекламы от компании Google, предоставляющий удобный интерфейс и множество инструментов для создания эффективных рекламных сообщений [4]. Сервис Google Adwords постепенно приближается к номеру 1 на просторах Рунета.

Основные характеристики сервисов контекстной рекламы Яндекс.Директ и Google AdWords представлены в таблице.

Таблица. Характеристики сервисов контекстной рекламы

Показатель	Яндекс.Директ	Google AdWords
Формирование рекламной кампании	Схема: компании, в них объявления, к которым относятся конкретные списки ключевых запросов	Структура: каждая из компаний разбивается на группы объявлений, которые содержат отдельные объявления, к ним относятся ключевые запросы
Интеграция с системами статистики	Яндекс.Метрика: система интуитивного понимания подачи данных выше	Google.Analitics: более сложная, гибкая и масштабная система настройки
Геотаргетинг	Возможность таргетироваться на большие города и области, а также на отдельно входящие в них районы	Отсутствуют многие небольшие города и районы. В некоторых случаях приходится вручную выбирать нужные места на карте. Есть возможность понижать/повышать ставки на определенный регион
Структура рекламного объявления	Заголовок, строка описания, целевой url, а также дополнительные ссылки. Общий объем знаков – 108, из них 33 – на заголовок	Заголовок, две строки описания по 38 символов в каждой, целевой и отображаемый url. Общий объем знаков – 106 символов. Необходимо тщательное структурирование объявления

Показатель	Яндекс.Директ	Google AdWords
Минус-слова	Преимущество: расставлять минус-слова можно и на уровне ключевых слов, что увеличит шансы показа созданного под запрос объявления	Главная проблема: плохое понимание системой склонения/спряжения слов, поэтому указывая минус-слова, следует добавлять все варианты слов
Рекламные сети	Обширная система контекстно-медийных сетей. Ограничены возможности работы с баннерными объявлениями	Обширная система контекстно-медийных сетей. Предоставляемые возможности работы с сетью намного шире
Аналитическая система подсказок поисковых запросов	Имеет ряд преимуществ: работа с конкретными цифрами, более удобный интерфейс	Имеет необходимый инструмент работы с подбором поисковых запросов
Ценовая политика	Оплата только за переходы (клики). Настройка стоимости клика	Возможность установить стоимость клика, а также бюджет расхода средств за день

Результаты анализа характеристик, представленных в таблице, позволяют сделать следующие выводы [5–7].

1. Обе системы преследуют цели максимально эффективно работать с рекламной компанией и дать возможность настроить контекстную рекламу так, чтобы получить от нее значительных результатов.
2. AdWords дает возможность настраивать рекламную компанию более тщательно, обладая обширным инструментарием для подгонки почти всех условий.
3. Яндекс.Директ более ориентирован на начинающего пользователя, не обладающего значительными познаниями в тонкостях настройки рекламной компании.
4. При подборе запросов, а также при настройке рекламной компании, обе системы эффективно работают как с широкими запросами, так и с фразовыми и с уточненными запросами.
5. По возможностям настроек геотаргетинга на территории России выигрывает Яндекс.Директ.

Литература

1. Бабаев А., Евдокимов Н. Контекстная реклама. – СПб.: Питер, 2011. – 304 с.
2. LiveInternet. Статистика сайта [Электронный ресурс]. – Режим доступа: <https://www.liveinternet.ru/stat/ru/searches.html?period=month;total=yes>, своб.
3. Wikipedia. Яндекс.Директ [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Яндекс.Директ>, своб.
4. Wikipedia. AdWords [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/AdWords>, своб.
5. Google AdWords [Электронный ресурс]. – Режим доступа: <https://www.google.ru/adwords/benefits/>, своб.
6. Яндекс [Электронный ресурс]. – Режим доступа: <https://direct.yandex.ru/>, своб.
7. Захарова И.В. Контекстная реклама // Технические науки – от теории к практике. – 2016. – № 57. – С. 64–70.



Логинов Максим Юрьевич

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа №
S4107с

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: akiam@mail.ru

УДК 004.514

ИССЛЕДОВАНИЕ СОВРЕМЕННЫХ ТЕНДЕНЦИЙ В UI/UX

Логинов М.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 рассмотрены понятия user experience/user interface, задачи, которые решают эти области, и рассмотрены некоторые примеры их современного использования.

Ключевые слова: UX/UI, юзабилити, интерфейс, user experience, user interface, тенденции в web-дизайне, web-дизайн, проектирование интерфейсов.

В настоящее время интерфейс как способ взаимодействия с пользователем является неотъемлемой частью стратегии развития интернет-проектов.

Дизайнерские идеи приводят к тенденции в веб-дизайне, одна приходит на смену другой довольно часто, как и версии библиотек для разработки веб-сайтов или же выход новых фреймворков. Позднее некоторые из трендов становятся стандартами.

От визуального представления зависит желание пользователя взаимодействовать с веб-сайтом, качество восприятия информации и конечный опыт. Основная мысль заключается в том, чтобы веб-сайт приобрел общую структуру, которая позволит наладить ментальную связь с пользователем, отделить элементы управления от значимой информации, тем самым оставляя взаимодействие понятными.

Проектирование дизайна, в свою очередь, заключается в оптимизации бизнес-процессов, которые происходят внутри компании, и удобства для пользователя, снимая с него нагрузку лишнюю нагрузку, такую как заполнение полей ввода данных или навигация по меню.

Из вышеописанного следует, что проблема общей доступности никогда не пропадет, и новые способы взаимодействия с конечным пользователем всегда актуальны, что позволяет улучшить показатели конверсии сайта [1].

Исследования в user experience (UX) не развивались из какого-то другого поля или полей. Это просто приемы из других видов исследований. Другими словами, UX позаимствовал многие техники исследования от академиков, ученых, исследователей рынка, и так далее. Тем не менее, все еще существуют некоторые методы исследования, которые являются достаточно уникальными для UX-мира.

Основной **целью исследования** являлся процесс проектирования с точки зрения конечного пользователя. Это исследование, которое позволяет охватить группу пользователей, а не одного конкретного, зачастую, разработчика. Это общепринятое правило, что целью UX или дизайна [2], направленного на пользователя, является разработка для конечного пользователя, и исследование, которое говорит нам, кто этот человек, и как он будет использовать этот продукт или услугу, и что им нужно от нас.

Имея это в виду, исследование состояло из двух частей: сбор данных, и обобщения этих данных для того, чтобы улучшить удобство и простоту использования. В начале работы, изучаются требования со стороны, которая предоставляет услугу, а также узнаются потребности и цели конечных пользователей. Обычно проводятся опросы предполагаемой целевой аудитории, текущих пользователей, а также производится анализ существующих данных и аналитики. Затем итеративно в процессе проектирования, исследования фокус смещается в сторону практичности и настроек. Исследователи могут проводить юзабилити тесты или А/В тестирование пользователей, и, как правило, проверки тезисов, которые позволят улучшить дизайн [3].

Можно выделить следующие тенденции в дизайне:

- контрастные цвета;
- большой размер шрифтов;
- svg-графика и анимации;
- стремление к минимализму и плоскому в дизайне.

Что касается пользовательского взаимодействия, то здесь можно наблюдать:

- короткие формы для заполнения данных, как правило, состоящие из нескольких полей;
- стремление к более интуитивному взаимодействию пользователя с сайтом.

Попробуем выделить преимущества каждого пункта:

- контрастные цвета помогают визуально выделить область, куда пользователь должны обратить свой взор в первую очередь, будь то кнопка для получения какой-либо услуги, или же текст, который явно его заинтересует [4];
- большой размер шрифтов обычно помогает выделить основную идею сайта, или определенной страницы;
- SVG-графика в вебе существует, можно сказать, продолжительное время, но не многие разработчики стремятся ее использовать, но возможность хранить только координаты и направления векторов, обычно занимает меньше памяти, по сравнению с растровой графикой, тем самым можно добиться более быстрой загрузки, или анимировать отдельные части картинки, придав странице динамичности. Не говоря уже о возможности масштабирования векторной графике без потерь [1];
- минимализм в дизайне пришел вместе с популярностью мобильных устройств, позволяющих выходить в интернет и просматривать сайты. С каждым годом посещение с мобильных устройств или планшетов возрастает, поэтому появилась необходимость оставлять на веб-страницах только самую важную информацию, ведь размеры экранов мобильных устройств в разы меньше настольных персональных компьютеров;
- короткие формы связи могут значительно повысить интерес пользователя оставить свои данные на сайте, ведь спросить по телефону, чем именно клиент заинтересовался, гораздо быстрее, чем заставлять его заполнять некоторое количество полей. Но здесь стоит учитывать бизнес-процессы компании и ее территориальную распределенность, но стандарт HTML5 может решить и эту проблему;
- и, наконец, «говорящие иконки», которые помогают пользователю, не читая текст, понять его смысл. Также «говорящие иконки» очень распространены в мобильных версиях сайта или приложениях, заменяя собой названия пунктов меню или текст кнопок функционала.

Таким образом, следование современным тенденциям может повысить посещаемость и конверсию на сайте [5].

Литература

1. Норман Д. Дизайн привычных вещей. – М.: Вильямс, 2006. – 384 с.

2. Complete Beginner's Guide to UX Research [Электронный ресурс]. – Режим доступа: <http://www.uxbooth.com/articles/complete-beginners-guide-to-design-research/>, своб.
3. Полное руководство новичка по UX исследованию [Электронный ресурс]. – Режим доступа: <http://uxgu.ru/ux-research/>, своб.
4. Маркотт И. Отзывчивый веб дизайн. – М.: Манн, Иванов и Фербер, 2012. – 163 с.
5. Уолтер А. Эмоциональный веб-дизайн. – М.: Манн, Иванов и Фербер, 2012. – 144 с.



Ломтев Илья Анатольевич

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: ilylomte7@gmail.com



Государев Илья Борисович

Год рождения: 1979

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

e-mail: ilia.gossoudarev@gmail.com

УДК 004

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ВЕБ-СИСТЕМ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ ЗАДАЧ ПО ПРОГРАММИРОВАНИЮ

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 рассмотрена важность электронного обучения, его преимущества и недостатки в сравнении с традиционным обучением, распространенные системы электронного обучения, их возможности.

Ключевые слова: дистанционное обучение (ДО), системы дистанционного обучения (СДО), электронное обучение (ЭО), Ideone.com, Onlinejudge, Timus Online Judge, Ejudge, Contester.

Электронное обучение (ЭО) [1] и дистанционные образовательные технологии (ДОТ) широко внедряются на всех уровнях образования, а их применение является перспективным для многих людей, включая тех, у которых нет времени или возможности на полноценное посещение учебного заведения, для людей, которые физически не могут посещать учебные заведения и т.п. Стираются пространственные рамки: не имеет значение местонахождение преподавателя и обучающегося. В Российской Федерации в статье 16, ФЗ-273 «Об образовании в РФ» нормативно закреплена возможность применения ЭО и ДОТ.

Так как программирование часто изучается в ходе приобретения практического опыта, количество выделенных часов в учебном заведении, содержание тестов и используемые технологии не являются достаточными для эффективного обучения программированию. Тесты преимущественно направлены на выбор ответа из

представленных, написание нужного ответа или установки соответствий. Но эти типы тестовых заданий не подходят для обучения программированию, так как необходимо проверять уровень сформированности навыков написания кода. Автоматизированные обучающие системы, например, системы дистанционного обучения (СДО), теоретически могут решить данную проблему, предоставляя обучающемуся возможность самостоятельно выбирать темп обучения, а преподавателю возможность разрабатывать различные типы тестов. В связи с этим была поставлена **цель**: изучить существующие в вебе системы автоматизированного тестирования, позволяющие проверять задачи по программированию.

СДО, помимо возможности самостоятельного выбора темпа обучения, обеспечивают свободу и гибкость обучения, его доступность вне зависимости от места проживания и социального статуса обучаемых; мобильность и технологичность. В то же время в работах [2] выделены проблемы, связанные с использованием ЭО и ДОТ. Наиболее важные из них – это отсутствие очного общения; сложность создания индивидуально-психологических условий; необходимость постоянного доступа к источнику информации; отсутствие универсальности (не по всем направлениям подготовки возможно эффективное удаленное обучение, а по некоторым направлениям подготовки возможна лишь реализация смешанной модели обучения). ДОТ используются и в подготовке будущих программистов. Однако традиционные подходы к разработке структуры учебных дистанционных курсов [2] оказываются недостаточно эффективными для разработки таких курсов, ориентированных на обучение программированию. Особое место в создании учебных дистанционных курсов занимает промежуточное и итоговое (по курсу в целом) тестирование. Как показал анализ учебных дистанционных курсов по программированию, в большинстве из них реализуется три основных типа тестов: выбор одного или нескольких ответов из представленных, ввод с клавиатуры нужного ответа или установка соответствий. Однако эти типы тестов не подходят для промежуточного и итогового тестирования сформированных компетенций у будущих программистов, так как многие навыки связаны с написанием кода. Аналогичная проблема видится и с разработкой практических заданий. Некоторые СДО имеют необходимый инструментарий для решения этой проблемы, т.е. предоставляют преподавателю возможность разрабатывать различные другие типы практических и тестовых заданий. С помощью подобных систем реализуют общую методологию разработки программного обеспечения (ПО) (TDD, test driven development), которая в аспекте образовательного процесса становится мотивированным тестом-обучением (TDI, test driven instruction) [3].

Рассмотрим возможности некоторых наиболее распространенных СДО применительно к их применению в обучении будущих программистов. Moodle (модульная объектно-ориентированная динамическая учебная среда) – это свободная система управления обучением. Она ориентирована на организацию взаимодействия между обучающим и обучающимся, а также подходит для организации традиционных дистанционных курсов. Данную систему можно установить на локальный сервер под управлением операционной системы (ОС) Windows или Linux. Она может быть использована для обучения программированию в связке с Ideone.com и Onlinejudge.

Ideone.com и Onlinejudge – две системы, которые дополняют друг друга и функционируют в симбиозе. Ideone.com – это онлайн-компилятор и инструмент для отладки, который позволяет компилировать и запускать код, записанный на более чем 60 языках программирования. Onlinejudge – это плагин для системы дистанционного обучения moodle, который дополняет стандартные типы заданий заданиями по программированию с ответом в виде написанной программы; с помощью этого плагина можно проверять и тестировать программный код на локальном сервере или на Ideone.com. Onlinejudge при размещении на локальном сервере под управлением ОС

Linux обладает полным контролем над приложениями, а под управлением ОС Windows – только частичным. Onlinejudge самостоятельно, без использования Ideone.com, поддерживает небольшое количество языков программирования, чтобы добавить языки программирования, нужно вручную редактировать файлы конфигурации.

Timus Online Judge – веб-ресурс, который разработали в Уральском федеральном университете. Он находится в свободном доступе и представляет собой большую базу задач по программированию и их решения на различных языках программирования. Разработчики утверждают, что это «крупнейший в России архив задач по программированию с автоматической проверяющей системой».

Ejudge – система автоматизированного тестирования отечественной разработки. Ее архитектура предусматривает ПО на сервере и заготовку интерфейса клиентского доступа через веб-интерфейс. Для предоставления пользователям доступа к системе администратору нужно самостоятельно разработать веб-ресурс в соответствии со стандартами Ejudge. Работы данной системы возможны только с использованием локального сервера с ОС Linux или Windows (рекомендуется Linux). У проекта есть обширная документация, в том числе и на русском языке. Ejudge поддерживает самые распространенные интерпретаторы и компиляторы (по умолчанию это – Pascal, C, C++, C#), для расширения их списка необходимо редактировать файлы конфигурации.

Contester – еще одна система автоматизированного тестирования отечественной разработки. Ее архитектура также предусматривает ПО на сервере и клиентский доступ через веб-интерфейс. Contester предназначен для использования на локальном сервере, с ОС Windows или Linux – обе версии имеют одинаковый функционал. Настроить систему можно через веб-интерфейс или отредактировав файлы конфигурации. Contester обладает большим перечнем различных «болванок» задач и готовых турниров. В документации проекта минимальное описание возможностей и функций системы, которое необходимо для пользования Contester. Список доступных для проверки языков программирования система определяет автоматически: сканируется система сервера или редактируются файлы конфигурации.

В данной работе представлены результаты анализа только небольшой части существующих на рынке образования СДО. Дополнительно проведенный анализ подтвердил, что большинство СДО обладают широким спектром функциональности, а их многообразие позволяет выбрать СДО, наиболее подходящую для решения поставленных образовательных задач. Однако результаты анализа также показали, что в настоящее время ни одна СДО не предоставляет такой возможности, как анализировать добавленные задачи для вычисления их рейтинга по уровню сложности, так и начислять баллы за их решение. Этот подход реализован в системе TopCoder для проведения международных соревнований по спортивному программированию. Эти соревнования проводятся в виртуальной среде при помощи специального приложения, имеющего свою систему пересчета «рейтинга» участников, после проведения очередного соревнования [4]. Данный подход можно использовать для большей автоматизации работы в СДО, а также в обучении программированию, так как преподавателю останется лишь разработать задания для проверки знаний, а их сложность определит сама система.

Литература

1. Государев И.Б. К вопросу о терминологии электронного обучения // Человек и образование. – 2015. – № 1(42). – С. 180–183.
2. Готская И.Б. О методике проектирования учебного дистанционного курса // Современное образование: традиции и инновации. – 2016. – № 4. – С. 24–29.
3. Государев И.Б., Жуков Н.Н. Мотивированное тестами обучение программированию // Современное образование: традиции и инновации. – 2016. – № 4. – С. 57–62.
4. TopCoder – Help [Электронный ресурс]. – Режим доступа: <https://community.topcoder.com/tc?module=Static&d1=help&d2=ratings>, своб.



Лунев Алексей Витальевич

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: lunev1905@gmail.com

УДК 004.92

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ИНСТРУМЕНТОВ ДИЗАЙНА И ПРОЕКТИРОВАНИЯ ИНТЕРФЕЙСОВ WEB-ПРИЛОЖЕНИЙ

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 исследований и выявлены основные требования к дизайну интерфейсов. Рассмотрены инструменты дизайна и проектирования интерфейсов web-приложений и их основные характеристики. Выделено несколько важных проблем выбора инструментов проектирования и дизайна. Поставлены актуальные задачи для последующих исследований.

Ключевые слова: web-дизайн, проектирование, прототипирование, интерфейс, web-сайт, web-приложения.

Актуальность работы обусловлена тем, что web-дизайн становится более универсальным инструментом контроля процесса информационного взаимодействия. Все чаще элементы и формы, вызывающие противоречивые мнения, сменяются осознанным расположением web-объектов, которые отвечают принципам эстетического информационного пространства. Таким образом, web-дизайн используется как инструмент формирования и содействия процесса взаимодействия человека и web-среды.

Рассмотрим возможности применения методов и инструментов дизайна web-интерфейсов для создания дизайна и прототипирования интерфейсов web-приложений. Web-сайт, процесс создания которого имеет свои особенности, можно представить как информационную систему. В настоящее время отдельные этапы создания web-сайта хорошо отлажены: маркетинг, юзабилити, web-программирование, тестирование, техподдержка. В то же время разработка дизайна интерфейсов web-сайтов не носит комплексного характера, отсутствует связь между данными этапами. В связи с этим весьма актуально проведение системных исследований, обращенных на выявление методов проектирования интерфейсов web-сайтов. Важной задачей является использование современных технологий и инструментов дизайна и прототипирования интерфейсов web-приложений для оптимизации и рационализации процесса разработки.

Анализ исследований [1–4] позволил выявить требования, которым должен соответствовать эффективный интерфейс сайта, а именно он должен быть:

- простым и интуитивно понятным;
- быстрым – нужная информация должна лежать в пределах одного-двух кликов;
- доступным – информация, которую должен увидеть пользователь, обязательно должна быть размещена на главной странице или на нее должна быть ссылка на самом видном месте;
- видимым – окно поиска, ссылки и различные модули должны быть хорошо видны (не сливаться с общим фоном и не прятаться за лишними панелями и т.д.);
- иметь простые имена – из названий разделов должно быть сразу понятно их содержание;
- просматриваемым – текстовая информация всегда должна быть структурирована по блокам;

– не слишком громоздким – это также относится к текстовому контенту.

Основные характеристики инструментов дизайна и проектирования интерфейсов web-приложений представлены в таблице.

Таблица. Основные характеристики инструментов дизайна и проектирования интерфейсов web-приложений

Инструмент	Описание
Lumzy	Инструмент для создания набросков web-сайтов и приложений. Имеет поддержку совместной работы над проектом, для этого имеет чат для обсуждения дизайна и набросков, указания версии файлов и многое другое.
Mockingbird	Онлайн-инструмент, упрощающий процесс создания интерфейса, с возможностью кооперироваться, присутствует возможность предварительного просмотра и обмена набросками через Интернет.
Mocklinkr	Имеет возможность разместить макет на Mocklinkr, вместо отправки по электронной почте, а заказчик получит ссылку. Можно оставлять комментарии на макете, создавать кликабельные области на макете, ссылки на другие макеты.
MockFlow	Онлайн-приложение для разработки набросков пользовательских интерфейсов для web-сайтов и ПО.
DENIM	Инструмент для разработки дизайнов пользовательского интерфейса web-сайтов на ранней стадии.
Hot Gloo	Онлайн-приложение для создания каркасов интерфейсов.
Axure RP Pro	Инструмент для создания прототипов web-сайтов. Генерирует кликабельный HTML и документацию в формате Word. Поддерживает совместное редактирование.
OverSite	Создает структуру приложения, позволяет проектировать интерфейсы и моделировать приложения в виде кликабельного прототипа. Существует возможность импорта существующего сайта для использования в качестве отправной точки.
UXPin	Онлайн-платформа для дизайна пользовательского взаимодействия и интерфейсов, прототипов десктопных, web- и мобильных приложений. Прототипы могут быть созданы из встроенных пресетов и библиотек, файлов Photoshop, импортированных файлов Sketch либо с использованием обширных библиотек пользовательских интерфейсов UXPin.
Sketch	Профессиональный векторный графический редактор, заточенный непосредственно под создание графических интерфейсов: web-, мобильных приложений, десктопных программ и пр.

Можно выделить несколько важных проблем выбора инструментов проектирования и дизайна. Нет единого инструмента, позволяющего конвертировать прототипы в полноценные макеты в рамках одной системы. Системы пресетов и шаблонов для работы с большими интерфейсами, в которых одни и те же элементы повторяются многократно, реализованы далеко не везде. Анимированные интерфейсы невозможно создать в большинстве инструментов. Имеется необходимость в универсальном инструменте, позволяющем не только проектировать, обрисовывать и анимировать в одной системе, но и даст возможность удобной связи и взаимодействия команды разработчиков или проектной группы на уровне обсуждений, комментирования и контроля версий без помощи других продуктов.

Проведенный анализ инструментов проектирования дизайна интерфейсов web-приложений, а также выделенные требования, которым должен соответствовать современный эффективный интерфейс сайта, позволили сформулировать ряд актуальных задач для последующих исследований [5–7]:

- анализ научно-технической информации в области существующих моделей проектирования и процесса дизайна, методов и подходов к дизайну, разработке интерфейсов web-сайтов;
- анализ процесса создания интерфейса web-сайтов;
- анализ и классификация инструментов для их разработки интерфейсов web-приложений;
- анализ особенностей современного web-дизайна.

Литература

1. Корзина М.И., Лысенко В.А., Лысенко А.А. Системное проектирование (дизайн) web-сайта // Дизайн. Материалы. Технология. – 2013. – № 1(26). – С. 116–122.
2. Корзина М.И., Костюченко О.А., Лысенко В.А. и др. Моделирование эстетического оформления веб-сайта // Вестник Северного (Арктического) федерального университета. – 2013. – № 1. – С. 116–123.
3. Корзина М.И., Хаймина Л.Э. Фестиваль «IT-Архангельск 2012» // Вестник Северного (Арктического) федерального университета. – 2013. – № 1. – С. 124–126.
4. Кирсанов Д. Веб-дизайн: книга Дмитрия Кирсанова. – СПб.: Символ-Плюс, 2007. – 368 с.
5. Писаренко Т.А., Ставнистый Н.Н. Основы дизайна: учебное пособие. – Владивосток: ТИДОТ ДВГУ, 2005. – 112 с.
6. Филенко Р.Е. Композиция в Web // Современное искусство: проблемы, тенденции, перспективы. – 2011. – С. 258–266.
7. Гэллоуэй А.Р. Неработающий интерфейс // Медиа: между магией и технологией. – 2014. – С. 252–288.



Магнитова Мария Владимировна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: magnitova@gmail.com



Государев Илья Борисович

Год рождения: 1979

Факультет методов и техники управления «Академия ЛИМТУ»,

кафедра компьютерного проектирования и дизайна, к.педагог.н., доцент

e-mail: ilia.gossoudarev@gmail.com

УДК 004

АНАЛИЗ СЕРВИСОВ ОПРЕДЕЛЕНИЯ ПОЗИЦИИ ВЕБ-РЕСУРСА В ПОИСКОВОЙ ВЫДАЧЕ

Магнитова М.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 проведен анализ существующих сервисов определения позиции веб-ресурса в выдаче поисковых систем Яндекс и Google по вводимым запросам, оценена корректность работы сервисов, проведено сравнение предлагаемого функционала сервисов, представлены структурированные результаты.

Ключевые слова: веб-ресурс, позиция веб-ресурса, поисковая выдача, поисковая система, сервис определения позиции, запросы, продвижение.

С каждым годом интернет все прочнее входит в нашу повседневную жизнь. Интернет-магазины и веб-ресурсы других коммерческих и некоммерческих организаций заинтересованы в том, чтобы находиться на первой странице, а точнее в первой тройке поисковой выдачи по актуальным для них запросам. Наличие позиции веб-ресурса в первой десятке или тройке поисковой выдачи увеличивает вероятность притока пользователей на данный веб-ресурс, что впоследствии увеличивает вероятность заказа товара, увеличение прибыли (или роста других качественных показателей организации).

Повышение позиции веб-ресурса в поисковой выдаче – это комплексная задача, которая относится как к маркетинговой деятельности организации, так и к уникальности и своевременности публикуемого контента, часто требует анализа структуры веб-ресурса и др.

Для понимания положения своего веб-ресурса среди конкурентов организации системно проводят оценку позиций по интересующим их запросам и на основе полученных данных корректируют свою стратегию развития. Определить позицию веб-ресурса в поисковой выдаче можно при использовании специализированных онлайн-сервисов или «ручным» методом. Последний отражает действительное положение веб-ресурса, но является более трудозатратным, особенно для организаций, находящихся далее чем на 20-й позиции поисковой выдачи, и при необходимости проверки более 5 запросов в более чем двух поисковых системах. Использование специализированных сервисов является быстрым и современным подходом, снижающим трудозатраты и предотвращающим ошибки человеческого фактора.

На сегодняшний день существует девять подобных сервисов: allpositions.ru, top-inspector.ru, a.pr-cy.ru, 3wseo.ru, siteposition.ru, webmasters.ru, sitexpert.org, www.seogadget.ru, sites-positions.ru. Из них только четыре удовлетворяют задаче данного исследования – определение позиции веб-ресурса в поисковой выдаче по вводимым запросам: allpositions.ru, top-inspector.ru, siteposition.ru, www.seogadget.ru [1].

Для анализа работы выбранных сервисов в качестве примера используется веб-ресурс коммерческой организации, являющейся представителем разработчика российской справочно-правовой системы на территории Санкт-Петербурга (СПб) и Ленинградской области (ЛО).

В качестве маркетинговых запросов в данном случае были определены следующие словосочетания:

- Название системы;
- [Название системы] купить;
- [Название системы] СПб;
- [Название системы] ЛО;
- [Название системы] купить СПб;
- [Название системы] купить ЛО [2].

Для аудитории Российской Федерации наиболее часто используемыми поисковыми системами являются поисковая система Яндекс и поисковая система Google, на которые и делается уклон при анализе сервисов оценки позиции веб-ресурса в данной работе [3].

Определение корректности информации, предоставляемой сервисом относительно рассматриваемого веб-ресурса по выделенным ранее запросам, производится в

сравнении с результатом, полученным «ручным» методом по каждому запросу в выбранных поисковых системах.

Все действия выполняются в браузере в режиме инкогнито для того, чтобы предпочтения пользовательского режима не повлияли на рассматриваемую поисковую выдачу, а также в один день в максимально короткий промежуток времени. При определении позиций поисковой выдачи в расчет не берутся рекламные позиции.

Рассматриваемые онлайн-сервисы определения позиции веб-ресурса в поисковой выдаче показали данные, равные или приближенные к данным «ручного» метода, по запросам в поисковой системе Яндекс, и только сервис allpositions.ru отобразил данные, ближе всех находящиеся к данным «ручного» метода в поисковой системе Google.

Результаты определения позиции веб-ресурса по выбранным запросам специализированными сервисами представлены в таблице.

Таблица. Позиции веб-ресурса по запросам в поисковой выдаче

	Запрос № 1	Запрос № 2	Запрос № 3	Запрос № 4	Запрос № 5	Запрос № 6
Поисковая система Яндекс						
Ручной метод	15	6	11	6	2	2
allpositions.ru	90	86	12	51	1	–
top-inspector.ru	–	–	–	–	–	–
siteposition.ru	17	5	11	27	2	1
www.seogadget.ru	21	6	13	8	1	1
Поисковая система Google						
Ручной метод	–	69	–	30	–	12
allpositions.ru	–	85	13	30	12	11
top-inspector.ru	–	–	–	–	–	–
siteposition.ru	90	12	11	21	11	7
www.seogadget.ru	–	–	–	–	11	9

Обнаружены технические неполадки в работе сервиса top-inspector.ru: при указанном времени ожидания – около 24 ч – сервис не предоставляет результаты более 48 ч.

Выявленные недостатки текущих сервисов определения позиции веб-ресурса в поисковой выдаче могут быть «отправной точкой» будущих разработок в данной области, а отраженные преимущества определенных сервисов могут упростить и одновременно сделать качественней работу интернет-маркетологов и деятельность организаций в целом [4].

Литература

1. Виноградов И.С. Сервис проверки позиций сайта в поисковых системах // Высшая школа. – 2016. – № 12. – С. 112–113.
2. Хронусова Т.В. Специфика поискового запроса в русскоязычном сегменте интернет // Российский технологический журнал. – 2015. – № 4. – С. 378–385.
3. Маматов Е.М. О продвижении сайта в поисковых системах Яндекс и Google // Научные ведомости Белгородского государственного университета. – 2014. – № 1(172). – С. 130–134.
4. Бочков Д.С. Факторы, влияющие на продвижение коммерческих сайтов в интернет-сети // Современное бизнес-пространство: актуальные проблемы и перспективы. – 2014. – № 1. – С. 129–132.



Маревская Мария Владимировна

Год рождения: 1989

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: marevskaia maria@gmail.com



Флеров Александр Викторович

Год рождения: 1956

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, ст. преподаватель

e-mail: kpd@limtu.ru

УДК 004.921

ИНФОГРАФИКА В ИНТЕРНЕТЕ: ОНЛАЙН-ИНСТРУМЕНТЫ

Маревская М.В., Флеров 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 области применения инфографики, ее определение, функции, критерии создания и классификация, онлайн-инструменты для создания инфографики, их особенности, преимущества использования.

Ключевые слова: инфографика, определение инфографики, классификация инфографики, онлайн-инструменты, шаблоны.

Информационная графика (инфографика) – предмет исследования специалистов таких областей знаний как искусствоведение, педагогика, культурология, журналистика, лингвистика. Единого толкования этого термина в научных исследованиях не существует.

По мнению Г.А. Никуловой и А.В. Подобных «главным признаком, позволяющим включить визуальный объект в подмножество информационной графики, является его способность представлять большой объем разнообразной информации в организованном виде, удобном для восприятия, или являться средством указания на действие или значение других видов информации». Инфографика выполняет иллюстративные, когнитивные и коммуникативные функции [1].

В работе [2] выделены следующие критерии создания информационной графики: актуальность, точность и объективность, вкус в вопросах дизайна, простота восприятия, краткость.

Существует несколько классификаций инфографики. С.В. Остриков выделяет дизайнерскую, стандартизованную и автоматическую инфографику [3]. Дизайнерская инфографика является результатом проектно-художественной деятельности, стандартизованная – создается на основе готовых шаблонов, автоматическая – генерируется программой.

Основную роль в развитие информационной графики вносит прогресс информационно-телекоммуникационных технологий: появляются новые виды

инфографики – видеоинфографика, интерактивная инфографика и др. В области веб-технологий – это осуществляется с помощью HTML5 Canvas и SVG, JavaScript. Существуют библиотеки и фреймворки JavaScript, а также большое количество онлайн-сервисов, более 50 [4].

Под инфографикой в онлайн-сервисах понимается таблица или график, сопровождаемый дополняющим его текстом. Под инфографикой в терминологии библиотек и фреймворков JavaScript понимается только график. В этом случае понятие информационной графики соприкасается с понятием визуализации данных.

Онлайн-инструменты, такие как Picktochart.com, Infogr.am, Easel.ly, Gliffy, Creately.com, Visual.ly, Venngage.com, Draw.io [4, 5], позволяют создавать стандартизованную инфографику и визуализировать данные.

В онлайн-сервисах используются наборы пиктограмм и символов, предтечей которых являются изотипы Отто Нейрата (1862–1945) [5], австрийского философа, социолога и экономиста.

К преимуществам онлайн-сервисов можно отнести: пояснения по элементам дизайна и композиции; результат адаптирован для web; современный, качественный, хотя и стандартизованный дизайн; автоматизация рутинных задач для дизайнера.

Благодаря интернет-технологиям происходит развитие инфографики, появляются новые ее виды, и она становится более доступной для пользователя. Онлайн-сервисы задают подход при составлении информационной графики. Они позволяют создать грамотную инфографику, отражают в себе теоретические концепции, продолжают традиции оформления, и адаптированы к современным требованиям размещения в вебе. Их можно широко применять в образовательных целях.

Литература

1. Никулова Г.А., Подобных А.В. Средства визуальной коммуникации – инфографика и метадианализ // Образовательные технологии и общество. – 2010. – № 2. – С. 369–387.
2. Беляев Н.А. Визуализация данных: инфографика как инструмент маркетинга // Вестник ЮУрГУ. Серия: Экономика и менеджмент. – 2015. – № 4. – С. 125–130.
3. Остриков С.В. Проектно-художественное моделирование инфографики: теоретические основы и принципы: автореф. дис. ... канд. искусствоведения. – М.: Моск. гос. худож.-пром. ун-т им. С.Г. Строганова, 2014.
4. Бекузарова Н.В., Ермолович Е.В., Ткачева А.В. Аналитический обзор сервисов инфографики // Современные проблемы науки и образования. – 2015. – № 6-0. – С. 400.
5. Грушевская В.Ю. Принципы использования онлайн-редакторов инфографики // Педагогическое образование в России. – 2015. – № 7. – С. 58–63.



Мартюшов Михаил Валентинович

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: martyushovmv@gmail.com



Готская Ирина Борисовна

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна,
д.педагог.н., профессор
e-mail: iringot@mail.ru

УДК 004.738.5

ПРОДВИЖЕНИЕ ВЕБ-САЙТОВ: ОСНОВНЫЕ МЕТОДЫ, ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ

Мартюшов М.В., Готская И.Б.

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

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе рассмотрено текущее состояние рынка услуг по продвижению веб-сайтов, определены основные тренды и проблемы данной сферы, и проведена оценка возможных перспектив.

Ключевые слова: продвижение, SEO, тренды, крауд-маркетинг, контент, оптимизация, SMM.

SEO (Search Engine Optimization) представляет собой комплекс мер по повышению позиции сайта в поисковых системах, и, таким образом, позволяет увеличить его целевую посещаемость [1]. Основной задачей SEO является привлечение как можно большего количества посетителей для последующего достижения определенных целей, которые ставят перед собой частные лица или компании. Поисковая оптимизация – это не просто процесс, а целая наука, цель которой увеличить популярность сайта. Причем важно не просто привлечь посетителя на сайт, но побудить его совершить определенное действие, например, купить товары или воспользоваться услугой с сайта.

За последнее время SEO значительно преобразилось и стало своего рода мощнейшим инструментом для достижения корпоративных целей и задач бизнеса. Если раньше данный инструмент расценивался как особая разновидность интернет-спама, то сейчас задачей SEO-оптимизаторов является достижение реальной авторитетности и популярности ресурса, а не искусственная имитация роста ссылочной массы.

Целью работы являлся анализ состояния рынка услуг по продвижению веб-сайтов, выделение основных трендов и проблем в данной сфере, а также анализ перспектив.

Основные методы продвижения сайтов. В настоящее время рынок услуг по продвижению сайтов перенасыщен предложениями. Современные SEO-агентства могут использовать в работе различные схемы взаимодействия с клиентами. Выделяют четыре основных способа:

- продвижение с оплатой за позицию. Это продвижение ресурса на первые страницы поисковой выдачи по конкретным запросам. Оплата производится за достижение определенной позиции в списке результатов поиска;
- продвижение с оплатой за лиды. Это вид продвижения, при котором на сайт привлекаются потенциальные клиенты. Оплата производится за целевое действие на сайте, выполненное привлеченным посетителем, либо за прямой контакт заказчика с привлеченным потенциальным клиентом;
- продвижение с оплатой за трафик. Это деятельность по привлечению новых пользователей на интернет-ресурс. Оплата производится за конкретные переходы пользователей на сайт из поисковых систем;

– продвижение с оплатой за работы. Это вид продвижения, при котором оговаривается ежемесячный перечень работ по поисковой оптимизации. Оплата производится исключительно за выполненные по факту работы.

Каждая из этих схем имеет свои достоинства и недостатки, и выбирать какую-либо из них нужно в соответствии с конкретной ситуацией.

С каждым новым обновлением поисковые системы становятся все «умнее», а люди, занимающиеся SEO-оптимизацией, уже не имеют такого большого влияния на выдачу сайтов в списках поисковиков, как год или два назад [2]. В связи с этим на первый план выходят другие методы продвижения.

1. Крауд-маркетинг. Среди новых методов продвижения, быстро набирающих популярность, однозначно выделяется крауд-маркетинг. Это комплекс мер, главной целью которых является создание положительного имиджа компании в интернете. Достигается это посредством размещения рекомендаций и положительных отзывов на разных сайтах, в том числе и в социальных сетях. Многочисленные исследования показывают, что большая часть пользователей в интернете перед совершением какой-либо покупки читают отзывы. Многие люди приобретают конкретный товар только потому, что увидели одобрительный пост об этом товаре в социальной сети у кого-то из списка своих друзей. Именно поэтому данный метод продвижения так популярен и эффективен.
2. SMM. Социальные сети являются одним из лучших источников поиска новых клиентов в интернете. В этом очень помогают сообщества по интересам. Именно там проще всего найти потенциальных клиентов и привлечь их внимание. Кроме того, в социальных сетях можно устраивать различные конкурсы, с помощью которых можно быстро добиться популярности.
3. Контекстная реклама. Реклама в интернете сейчас присутствует практически на каждой странице. Все большее распространение получает контекстная реклама, содержание которой напрямую зависит от интересов конкретного пользователя. Это очень мощный инструмент продвижения. Использование такой рекламы в социальных сетях и мобильных приложениях гарантированно увеличивает поток посетителей на сайт.
4. E-mail-маркетинг. Рассылка писем с рекламой не самый удачный способ привлечения новых пользователей. Очень часто такие письма блокируются антиспам-системами электронной почты или вызывают у получателей негативные эмоции. Но, если пользоваться этим способом грамотно и с умом организовывать рассылку, то можно получить немало новых потенциальных клиентов, которые впоследствии могут стать постоянными посетителями сайта.

Эти методы не гарантируют стопроцентный успех от их применения, но позволяют по максимуму распределить время, выделенное на продвижение. При этом можно легко проанализировать, что было сделано за конкретный промежуток времени и, ориентируясь на это, планировать дальнейшие действия.

Проблемы и перспективы. Основной проблемой в сфере продвижения сайтов является нестабильность и изменчивость обстановки. Поисковые системы постоянно совершенствуются и изменяют свои алгоритмы, тем самым выдвигая одни методы продвижения на передний план и уменьшая при этом важность других. Вследствие этого, оптимальная формула продвижения постоянно меняется. В таких условиях всегда существует риск потерять не только вложенные в продвижение средства, но и драгоценное время.

Актуальная формула эффективного продвижения ресурса в сети интернет выглядит так: SEO (одна из четырех схем взаимодействия) + E-mail-маркетинг + Crowd marketing + Контекстная реклама + SMM. Какой она будет через полгода или год предсказать очень сложно. Однако в последнее время прослеживается четкая тенденция увеличения роли

социальных сетей, видеоресурсов и мобильных приложений в продвижении сайтов. Именно эти направления стоит рассматривать в первую очередь, поскольку у них однозначно есть перспективы.

Вектор развития SEO все больше смещается в социальные сети. В связи с этим залогом успеха в продвижении веб-сайтов выступает использование комплексного подхода, важными компонентами которого являются такие методы как контекстная реклама, SMM и крауд-маркетинг. Появляется все больше сервисов, предоставляющих полезные услуги по применению данных методов продвижения в интернете. Есть большая вероятность, что именно за этими сервисами, будущее SEO-индустрии. В любом случае их актуальность в ближайшее время точно не уменьшится [3–5].

Литература

1. Бабаев А., Евдокимов Н. и др. Раскрутка. Секреты эффективного продвижения сайтов. – СПб.: Питер, 2013. – 215 с.
2. Смагина Н.В. Поисковое продвижение сайта: тенденции современного SEO // Ростовский научный журнал. – 2016. – № 7. – С. 13–20.
3. Романенкова О.Н. SEO-продвижение как эффективный инструмент интернет-маркетинга // Стратегии бизнеса. – 2014. – № 1. – С. 120–124.
4. Савельев Д.В., Симонов А.Б. Основные ошибки в продвижении сайтов // Актуальные вопросы экономического развития регионов. Сб. материалов V Всероссийской заочной научно-практической конференции. – 2016. – С. 30–32.
5. 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(19-1). – С. 372–375.



Матушевский Никита Евгеньевич

Год рождения: 1995

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4108

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: nickmat95@yandex.ru

УДК 005

ОПРЕДЕЛЕНИЕ И АНАЛИЗ НЕДОСТАТКОВ СУЩЕСТВУЮЩИХ МЕДИЦИНСКИХ ИНФОРМАЦИОННЫХ СИСТЕМ

Матушевский Н.Е.

Научный руководитель – д.т.н., доцент Погорелов В.И.

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

В работе приведены общие сведения, касательно медицинских информационных систем, приведен обзор наиболее распространенных проблем современных медицинских информационных систем, а также рассмотрена причина их появления и попытка ее решения. В заключение предложен вариант решения одной из проблем, включающий первичный план его реализации.

Ключевые слова: медицинская информационная система, МИС, проблемы медицинских информационных систем, проблемы МИС, дубликат карточки пациента.

Автоматизация информационных процессов в медицинской сфере началась в шестидесятых годах двадцатого века в США и западной Европе. Предпосылкой тому, стала нарастающая необходимость автоматизации процессов формирования и выставления счетов, оптимизация расчета стоимости услуг, а также увеличение объема информации, требующей обработки [1]. Удовлетворить эту необходимость удалось, путем создания медицинской информационной системы (МИС). МИС обеспечивает сбор, формирование, обработку, передачу, хранение электронных персональных медицинских записей, электронных историй болезни, финансовой, нормативно-справочной и вспомогательной информации [2]. Целью любой МИС является облегчение и упорядочивание работы с потоками медицинской информации. МИС позволяют повысить качество и скорость работы в условиях массового обслуживания пациентов. Достигается это путем автоматизации многочисленных глобальных процессов, таких как: документооборот, контроль затрат, заведение карточки пациента, возможность оценки эффективности лечения пациента, планирование складских запасов, контроль расхода лекарственных средств, расходных материалов на пациента и прочих.

Отечественные МИС классифицируются на [2]:

- медико-технологические информационной системы (ИС), предназначенные для информационного обеспечения процессов диагностики, лечения, реабилитации и профилактики пациентов в лечебно-профилактических учреждениях;
- информационно-справочные системы, содержащие банки медицинской информации для информационного обслуживания медицинских учреждений и служб управления здравоохранением;
- статистические ИС органов управления здравоохранением;
- научно-исследовательские ИС, предназначенные для информационного обеспечения медицинских исследований в клинических, научно-исследовательских институтах;
- обучающие ИС, предназначенные для информационного обеспечения процессов обучения в медицинских учебных заведениях.

Немалый вклад в исследование МИС внесли российские ученые (А.В. Гусев, А.И. Михеев, Я.И. Гулиев, И.Ф. Гулиева, О.А. Фохт, Е.В. Рюмина, В.Л. Малых), занимающиеся сбором статистики, обзорами, изучением проблем и поиском вариантов их решения, наблюдением изменений, и изучением перспектив развития МИС.

Несмотря на приблизительно одинаковый период появления, МИС в России значительно отстают от своих западных аналогов, процесс отставания начался еще в семидесятых годах и, в дальнейшем, лишь усиливался.

По опросу разработчиков медицинских информационных систем за 2012 год выяснилось, что лишь 7% разработчиков оценивают рынок МИС полностью сформированным, 50% находят его в стадии активного формирования, 29% считают, что рынок находится в зачаточном состоянии, еще 7% видят полное отсутствие рынка, и еще 7% говорят не о развитии рынка, а о его деградации [3].

Низкий уровень автоматизации медицинских учреждений и общий скепсис разработчиков являются следствием того, что за все годы существования и развития МИС в России до сих не сложилось единого понятия, о том, что такое МИС. У большинства компаний-разработчиков имеется свое понимание относительно того, что такое МИС, какие функции она должна выполнять и на чем стоит больше акцентировать внимание при ее создании. Терминологическая путаница, отсутствие единых критериев эффективности и требований к разработке значительно замедляет развитие МИС в России и усиливает отставание от европейских стран. На сегодняшний день данная проблема является основной в этой области [1].

Данная проблема способствовала появлению следующего пула проблем.

1. Сложности интеграции со сторонними информационными системами. Немаловажным

недостатком современных МИС является проблема, появившаяся еще в конце восьмидесятых годов и не решенная до сих пор – сложность интеграции со сторонними системами, например, с лабораторной ИС. В современной практике достаточно часто имеется необходимость взаимодействия сервисов для экспорта/импорта каких-либо данных в определенной последовательности, формате, количестве, из-за отсутствия единого стандарта обмена данными между программами в России. Процесс интеграции данных из одной системы в другую, производится либо вручную, что не является оптимальным, либо, путем доработки самим ИС, что также нельзя назвать оптимальным по ряду причин.

В Европе данная проблема была решена путем использования единого мирового стандарта HL7 – стандарт обмена, управления и интеграции электронной медицинской информации.

Данный стандарт поддерживает: структурирование передаваемых данных; возможности проектирования систем; достижение согласованности передач; безопасность; идентификация участников; доступность.

2. Трудности с наполнением системы. В современных МИС имеются трудности с наполнением системы, например, с импортом преискуранта или с импортом данных о пациенте. Основная проблема заключается в том, что данные должны быть представлены в определенном формате. Существуют два основных варианта импорта данных в систему: ввод данных вручную в форму пользовательского интерфейса или загрузка файла с данными также через пользовательский интерфейс.

В среднем необходимо импортировать около 1 000 строк в базу данных, что никак нельзя назвать оптимальным при ручном вводе.

Проблема загрузки файла заключается в его подготовке, собственно, в нужный вид файл приводится вручную, что также занимает достаточное количество времени, принимая во внимание человеческий фактор, в процессе подготовки может быть допущена ошибка, и данные импортируются некорректно. Очевидно, данный метод оптимальным назвать нельзя.

3. Наличие дубликатов карточек пациентов. Неразрешенной также осталось проблема наличия дубликатов карточек пациентов.

Дубликат карточки пациента – повторное заведение карточки на пациента.

Создание дубликатов карточек происходит исключительно из-за наличия человеческого фактора, чаще всего, это происходит при записи пациента на прием по телефону. Оператор, неправильно услышав фамилию и не найдя такого человека в системе, заводит новую карточку и отправляет ее врачу пациента, в связи с чем теряется целостность данных об истории болезни, что может повлечь за собой не только некачественное лечение пациента, но и нанесение вреда для его здоровья.

На данном этапе отсутствуют какие-либо требования или стандарты, решающие эту проблему. Отсутствует также и общее программное решение, позволяющее находить и объединять дубликаты карточек или препятствовать их созданию. Проблема решается либо путем ручного поиска дубликатов, либо путем создания частного решения. Однако действительно качественные частные решения присутствуют далеко не во всех МИС.

Попытка решения всех вышеперечисленных проблем была предпринята государством в 2008 году, путем создания единой и бесплатной МИС. Система получила название Федеральная типовая медицинская информационная система (ФТМИС).

Но данный вариант решения так и не получил признания в России. Идея бесплатной и единственной системы была воспринята профессионалами негативно. Аргументировано это было полным уничтожением конкуренции и монополизации рынка государством [1].

В результате исследования, удалось выделить четыре основные проблемы, а именно:

1. отсутствие единого понимания того, что такое МИС;
2. интеграция МИС со сторонними системами;
3. наличие дубликатов карточек пациентов;
4. сложности импорта данных в систему.

В проводимом исследовании предложено для решения проблемы наличия дубликатов карточек создание интегрируемого web-приложения, автоматизирующего процесс поиска и объединения дубликатов карточек в МИС. На вход в приложение поступает база данных. После проведения поиска дубликатов пользователю выводятся все возможные дубликаты карточек с процентом совпадения данных. После проверки пользователь выбирает реальные дубликаты карточек, а также карточку, с которой будет производиться сведение и нажимает кнопку сведения карточек, после чего приложение объединяет все выбранные карточки в одну. Алгоритм поиска дубликатов будет составлен на основании алгоритма нечеткого поиска данных, а также алгоритма поиска по определенным параметрам, например, ФИО, дата рождения, e-mail и так далее.

Нечеткий поиск – это поиск информации, при котором выполняется сопоставление информации заданному образцу поиска или близкому к этому образцу значению. Алгоритмы нечеткого поиска используются в большинстве современных поисковых систем, например, для проверки орфографии. Использование нечеткого поиска в реализации алгоритма поиска дубликатов позволит обнаружить фамилию, имя или отчество, написанное с ошибкой.

В дальнейшем предполагается использовать связки из параметров. Например, ФИО + дата рождения, ФИО + дата рождения + номер паспорта.

Разрабатываемое приложение позволит значительно увеличить эффективность поиска дубликатов, а также уменьшить время их поиска, и будет актуальным в первую очередь для небольших и молодых компаний-разработчиков, не реализовавших собственное решение.

Литература

1. Автоматизация учреждений здравоохранения РФ: текущий уровень, проблемы, тенденции [Электронный ресурс]. – Режим доступа: <http://cyberleninka.ru/article/n/avtomatizatsiya-uchrezhdeniy-zdravoohraneniya-rf-tekuschiy-uroven-problemy-tendentsii>, своб.
2. Медицинские информационные системы РФ [Электронный ресурс]. – Режим доступа: <http://cyberleninka.ru/article/n/meditsinskie-informatsionnye-sistemy-rf>, своб.
3. Рынок медицинских информационных систем: обзор, изменения, тренды [Электронный ресурс]. – Режим доступа: <http://cyberleninka.ru/article/n/rynok-meditsinskih-informatsionnyh-sistem-obzor-izmeneniya-trendy>, своб.



Микитенко Анастасия Сергеевна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4108

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: botasyamaus@gmail.com



Готская Ирина Борисовна

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна,
д.педагог.н., профессор
e-mail: iringot@mail.ru

УДК №004.912

ИССЛЕДОВАНИЕ МЕТОДОВ И ПРИЕМОВ ТИПОГРАФИКИ В ВЕБЕ

Микитенко А.С., 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».

В работе рассмотрены различные приемы типографики для веба. Подробно рассмотрены вопросы удобства использования тех или иных приемов в типографике. Приведены основные ошибки, совершаемые веб-дизайнерами в типографике. Рассмотрены вопросы шрифтов, начертания, выравнивания текста, работы со ссылками, выделения текста цветом и начертанием, употребление кавычек и тире, верстки, выравнивания текста по ширине и междустрочного интервала. Даются примеры и способы употребления типографики.

Ключевые слова: типографика, веб-типографика, шрифт, CSS.

Главным понятием печатной и экранной типографики является шрифт. Типографика играет важную роль в веб-дизайне, текст на сайте по статистике составляет около 95% [1]. Шрифтовое оформление управляет настроением и создает определенную атмосферу при прочтении текстового содержимого веб-страниц. Современная веб-типографика базируется на CSS-стилях. Меняя значения стилей браузера по умолчанию, можно сделать текстовое содержимое более привлекательным.

В веб-типографике применяются следующие основные принципы.

- Тип шрифта – это основа типографики. Выбор типа шрифта является ключевым этапом в разработке дизайна сайта.
- Группы шрифтов существуют с засечками и без засечек. На рисунке изображена востребованность шрифтов с засечками и без засечек.

Размер шрифта. Простой способ управления экранным текстом. Размер шрифта определяет иерархию размеров заголовков. В последнее время наблюдается стойкая тенденция к увеличению размеров шрифтов на всех сайтах [2]. Это связано в первую очередь с тем, что большой размер шрифта позволяет привлечь внимание пользователя к содержимому страницы, а также значительно улучшает восприятие.

Согласно исследованиям – наиболее распространенным является размер шрифта в 14–16 пикселей [3].

Все шрифты можно разделить условно на две категории:

1. шрифты, которые без проблем отображаются у большинства пользователей;
2. шрифты, которые отсутствуют у большинства пользователей.

До появления каскадных таблиц стилей CSS – проблема с отсутствующими шрифтами была существенной, и в основном веб-дизайнерам приходилось использовать стандартные шрифты в оформлении веб-страниц, либо использовать различные ухищрения, такие как использование технологии Flash, которая является шрифтонезависимой.



Рисунок. Шрифты с засечками против шрифтов без засечек

С появлением новых технологий это не является проблемой, достаточно прописать определенную конструкцию в таблицах стилей, и необходимый шрифт будет автоматически скачан и установлен на компьютер пользователя. Тем не менее, следует отметить, что это может приводить к дополнительному расходу трафика и уменьшению производительности, поэтому в большинстве случаев следует использовать типовые шрифты. Кроме того, при использовании загрузки веб-шрифтов документ может недостаточно корректно отображаться в разных браузерах.

– Акцент. Ключевые слова можно выделить следующими способами:

1. жирным;
2. наклонным;
3. подчеркнутым;
4. цветом;
5. большими буквами.

Большинство этих начертаний и способов выделения встроено в стандартные HTML-теги страниц, например, текст полужирного начертания можно получить, выделив нужный текст тегом `<i>`, а акцентировать внимание на определенном блоке текста можно с помощью тега ``. Все параметры этих тегов можно настроить в стилях CSS-документа.

- Расстояние между строками. Оптимальным расстоянием считается расстояние, не меньшее, чем один символ. Большое расстояние негативно сказывается на восприятии текста. Вертикальная дистанция между строк оказывает огромное влияние на стиль и четкость отображения текста.
- Расстояние между буквами. При изменении расстояния между буквами нужно учитывать выбранный шрифт.
- Расстояние между словами. CSS позволяет изменять расстояние между символами. Если уменьшать или увеличивать интервал, нужно обязательно предусмотреть, как это выглядит в различных браузерах. Также нельзя игнорировать следующие факторы: тип шрифта, расстояние между символами в слове.
- Границы и выравнивание. Есть несколько типов выравнивания – по левому краю, по правому краю, по центру и по ширине. Использование границ и выравнивания придает тексту более эстетический вид. Следует отметить, что использование выравнивания по ширине является «дурным тоном» в оформлении веб-страниц, поскольку это может приводить к некорректному отображению текста, например появлению неадекватных пробелов между словами и висячих строк из одного слова. Стандартом в типографике является использование выравнивания по левому краю, по последним исследованиям этот прием используется в 99% сайтов [3].

Средняя длина строки на современных сайтах составляет 570 пикселей и не меняется уже достаточно длительное время [3]. Это кажется несколько нелепым, учитывая, что разрешение мониторов постоянно увеличивается, и, казалось бы – на них уже можно размещать больше информации. Тем не менее, большинство сайтов используют стандартную длину строки, и связано это с тем, что большинство пользователей либо используют старые мониторы, либо выставляют на них низкое разрешение, поскольку это кажется им удобным. Повышение среднего размера строки следует ожидать через некоторое время, когда широкоформатные мониторы с высоким разрешением получат более широкое распространение.

- Выделение цветом используется на веб-сайтах достаточно редко. На большинстве сайтов используется стандартный темный текст на светлом фоне [4]. Использование различных ярких приемов и стилей является оправданным только для определенных сайтов, например, сайтов, предназначенных для детей. Также использование высококонтрастных схем оправдано при разработке сайтов для слабовидящих людей.
- Выделение ссылок является обязательным условием для любой веб-страницы. Следует отметить, что выделение ссылок стандартным синим цветом и подчеркиванием является стандартом, и должно использоваться в большинстве случаев. Использование каких-либо других методов может дезориентировать пользователя и значительно ухудшать восприятие веб-страниц.
- Верстка также является достаточно важным элементом типографики. В последнее время все более широкое распространение получает так называемая «резиновая» верстка, при которой веб-страница при изменении размера окна браузера – «подстраивается» под него. Это позволяет достигнуть хорошего вида страницы при любом размере и разрешении монитора у пользователя. Тем не менее, следует отметить, что использование данного приема не всегда оправдано, поскольку в некоторых случаях можно добиться обратного эффекта – ухудшить внешний вид страницы, и неправильно расположить элементы на ней. Особенно это актуально для низких разрешений.
- Кавычки и тире являются проблемой для многих веб-дизайнеров. В частности, многие из них не используют знак «длинное тире» и кавычки-елочки, принятые как стандарт в российской типографике. Это связано с тем, что использование дополнительных символов затрудняет набор веб-страниц. Тем не менее, следует отметить, что использование длинного тире и кавычек-елочек для оформления страницы является неотъемлемой частью хорошей типографики. И если вручную править типографику страницы в соответствии с правилами затруднительно – возможно использовать различные сервисы в сети Интернет, предназначенные для этого.

Проведенное исследование позволяет сделать вывод, что типографика является неотъемлемой частью дизайна веб-страниц. Хорошая типографика задерживает читателя на странице, позволяя максимально эффективно донести информацию. Использование нестандартных приемов в типографике в большинстве случаев не является оправданным, либо оправдано в исключительных случаях. Чем более «привычно» для пользователя выглядит страница, тем лучше она будет восприниматься им.

Литература

1. Петрова В.В., Калугина Д.А. Применение типографики для подготовки контента // Вестник МГУП им. Ивана Федорова. – 2015. – № 5. – С. 19–20.
2. Чихольд Я. Новая типографика. Руководство для современного дизайнера. – М.: Изд-во студии Артемия Лебедева, 2016. – 244 с.
3. Приемы и методы типографики в вебе [Электронный ресурс]. – Режим доступа: <http://fonts.jofo.me/239427.html>, своб.
4. Современные методы и приемы в типографике [Электронный ресурс]. – Режим доступа: <http://frontender.info/typographic-design-patterns-practices-case-study-2013/>, своб.

**Митропольская Анастасия Максимовна**

Год рождения: 1994

Факультет информационных технологий и программирования,
кафедра информационных систем, группа № М4205Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: nastis.mi@gmail.com

УДК 004.622

**МОДИФИКАЦИЯ ПРОГРАММНОЙ КОМПОНЕНТЫ SNAP DATASET
ПРОДУКТА MFE TIMEFINDER CLONE ДЛЯ ОБЕСПЕЧЕНИЯ РЕПЛИКАЦИИ
МНОГОТОМНЫХ НАБОРОВ ДАННЫХ С ТИПОМ ДОСТУПА VSAM****Митропольская А.М.****Научный руководитель – к.педагог.н., доцент Маятин А.В.**

Сегодня в век компьютеризации и информатизации информация и данные любого объема и степени важности подвержены постоянным угрозам как со стороны преднамеренной их порчи, так и со стороны неподконтрольных человеку происшествий. Репликация как целых дисков, так и отдельных наборов данных в рамках систем хранения данных (СХД) на уровне дискового массива обеспечивает безопасность данных. В работе рассмотрены решения нескольких проблем, возникших при проектировании репликации многотомных наборов данных.

Ключевые слова: системы хранения данных, репликация данных, наборы данных, Symmetrix, TimeFinder Clone.

Сегодня риски, связанные с недоступностью данных или их потерей, резко возросли, поэтому обеспечение надежного хранилища данных, способного гарантировать их защищенность, является одной из актуальнейших задач XXI века. Системы хранения данных Symmetrix предоставляют технологию репликации наборов данных. Управление этой операцией осуществляется с помощью программного обеспечения MFE TimeFinder Clone с мейнфрейма, работающего под управлением операционной системы z/OS. Наборы данных, располагающиеся на нескольких дисках, называются многотомными. Репликация многотомных наборов данных – очень важный функционал, поскольку копировать целиком все диски, на которых располагается набор данных, нецелесообразно, а в большинстве случаев набор данных является именно многотомным. В данной работе были рассмотрены решения нескольких проблем, возникших при проектировании функционала репликации многотомных наборов данных, а также спроектирован и разработан новый тип журналирования.

1. Выбор диска для размещения целевого набора данных. Пользователь указывает список из дисков, на которые можно записать набор данных. Продукт проверяет каждый диск – если он в статусе «онлайн», имеет аналогичный, исходному тип архитектуры, находится на той же систем хранения данных (СХД) и на нем достаточно свободного места, то диск добавляется в список. Затем список сортируется по количеству свободного места на диске, после чего набор данных записывается на диски с наибольшим количеством свободного места. Однако, если в списке не нашлось подходящих дисков, то продукт должен попытаться самостоятельно найти диск. В операционной системе z/OS каждому диску присваивается Storage Class и Management Class [1]. Storage Class – это набор дисков с одинаковыми техническими характеристиками. Management Class – это набор дисков с одинаковыми атрибутами управления. Таким образом, продукт ищет подходящие диски в соответствующих исходному диску Storage Class и Management Class. Поскольку диски в классах схожи по характеристикам, то фактически задача поиска сводится к поиску дисков с достаточным количеством свободного места.

2. Оптимизация ресурсов при копировании больших наборов данных, расположенных на множестве различных дисков. Наборы данных состоят из блоков, в каждом блоке целое количество дорожек. При репликации для целевого набора выделяется столько же блоков, сколько их есть у исходного. Таким образом, после репликации получится несколько пар дисков (исходный – целевой), на каждом целевом выделено столько же блоков, сколько выделено у соответствующего исходного диска. Это неэффективно, во-первых, с точки зрения ресурсов (велика вероятность ситуации, когда на нескольких дисках располагается по небольшому количеству дорожек), а во-вторых – с точки зрения производительности, поскольку придется искать такое же количество дисков. Помимо этого, существует вероятность того, что не найдется нужного количества подходящих дисков и репликация не осуществится, но при этом, если объединить все части исходного набора данных в одну, то его с легкостью можно будет разместить на количестве дисков, гораздо меньшем исходного. Решением этой проблемы является реализация параметра `ALLOC_EXTENT(YES|CONS_VOL|CONS_ALL)`. В соответствии со значением этого параметра определяется, можно ли менять размер и количество блоков данных исходного набора данных. Если указано значение параметра `YES`, то на целевых дисках выделяется такое же количество блоков данных такого же размера, как на исходном наборе. Второе возможное значение параметра – `CONS_VOL`, которое означает, что на каждом целевом диске будет размещено столько же дорожек, сколько занимает исходный набор данных на соответствующем исходном диске, а размер и количество блоков на одном диске могут изменяться. Последний вариант – `CONS_ALL`, который означает, что суммарное количество дорожек на исходных дисках равно выделенному суммарному количеству дорожек на целевых дисках, при этом распределение дорожек по целевым дискам может отличаться от исходных.
3. Ограничение количества операций, одновременно выполняемых над дорожками. На данный момент это число ограничено числом 4, т.е., например, одновременно реплицировать набор данных можно только на 4 целевых диска. При попытке запустить 5-ю репликацию происходит ошибка. Подобное поведение не очень дружелюбно для пользователя, так как он вынужден перезапускать скрипты, пока не завершится одна из операций. Увеличение количества одновременно возможных операций на данный момент невозможно. Решением этой проблемы стала организация поведения, аналогичная системам массового обслуживания с ожиданием [2]: как только максимальное число операций достигнуто, все последующие операции становятся в очередь в ожидание. С определенной периодичностью продукт проверяет, не завершилась ли какая-то из операций. Подобное поведение так непринудительно и регулируется параметром `WAITFORSESSION`, которому можно указать, сколько минут операция будет ждать в очереди. По истечении этого времени, если ни одна из активных операций не завершилась, ожидающая операция завершается с ошибкой.
4. Для контроля хода и результата операций используется журналирование. На данный момент продукт `MFE TimeFinder Clone` предоставляет несколько типов журналирования, в том числе базовое и полное отладочное. Отладочное журналирование неэффективно по производительности и затрудняет чтение журнала. Однако, в случае возникновения ошибки у пользователя, разработчикам нужна вся диагностическая информация об операции, чтобы правильно диагностировать и устранить проблему. Это отнимает много времени как у пользователя, которому приходится снова воспроизводить ошибку, так и у разработчиков. Чтобы устранить эту проблему, был реализован новый механизм журналирования: в ходе выполнения операции в фоновом режиме собирается вся необходимая диагностическая информация, а выводится только в случае ошибок и аварийных завершений. Это помогает сэкономить время пользователя и время устранения ошибки.

В данной работе были проанализированы поставленные проблемы, требующие исследования и поиска пути решения, и представлены возможные решения проблем. В каждой проблеме так или иначе вставал вопрос дружелюбности решения пользователю, поэтому

большинство решений подразумевают собой возможность выбора поведения продукта. По умолчанию во всех сценариях будет сделан упор на сохранение данных в том же виде, как они существуют на исходных дисках. Помимо этого, был реализован новый механизм журналирования, позволяющий уменьшить время устранения ошибки, полученной пользователем.

Литература

1. Managing Data with SMS [Электронный ресурс]. – Режим доступа: https://www.ibm.com/support/knowledgecenter/SSLTBW_2.1.0/com.ibm.zos.v2r1.idai600/i6037.htm, своб.
2. Романцев В.В. Аналитические модели систем массового обслуживания – СПб.: СПбГЭТУ (ЛЭТИ), 1998. – 67 с.



Мустафина Ляйсан Ирековна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4108

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: amix93@mail.ru



Дмитриевская Алёна Александровна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: dmitalyona@yandex.ru

УДК 004

АНИМАЦИИ В ВЕБ-ДИЗАЙНЕ: СОВРЕМЕННОЕ СОСТОЯНИЕ, ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ

Мустафина Л.И., Дмитриевская 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».

Анимация широко применяется в качестве инструмента, позволяющего представить материалы сайта с новой стороны. В работе рассмотрены популярные технологии анимирования сайтов, условия их применения и особенности веб-анимации.

Ключевые слова: анимация, веб-дизайн, CSS, JavaScript.

Понятие анимации в веб-дизайне едва ли можно назвать новым. Методы анимации отдельных элементов на сайте различной сложности входят в практику большинства разработчиков. Всплывающие баннеры, движущиеся логотипы, прокрутки на странице, раскрывающиеся формы стали привычными для современных пользователей. Однако изменилась основная идея в использовании анимации.

Применение анимации в веб-дизайне началось с повсеместного использования gif-файлов состоящих из множества картинок, обеспечивающих быстрой сменой эффект движения на изображении. Такая анимация использовалась только ради украшения и не несла в себе большой смысловой нагрузки.

С появлением технологии Flash стало возможно создание анимации более высокого уровня. Технология позволяла создать плавную анимацию, использовать видео- и аудиовставки. Однако необходимость использования сторонней программы Flash-player при просмотре, сильная загрузка ресурсов компьютера, найденный ряд уязвимостей привели к постепенному отказу от технологии и переходу к более современным методам анимации [1].

Сегодня анимация используется не столько для украшения сайта, сколько для улучшения понимания интерфейса пользователем. Например: анимация отдельных частей сайта для направления по заданному пути, приводящему к желанию пользователя совершить покупку.

На этом этапе наиболее популярные методы создания веб-анимации связаны с использованием технологий CSS и JavaScript, так как они дают широкие возможности для веб-дизайна. В таблице представлены особенности использования каждой из технологий.

Таблица. Особенности использования технологий CSS и JavaScript

№	CSS	JavaScript
1	Используется для простых, коротких переходов, таких как переключение состояния элементов пользовательского интерфейса	Используется для сложных сценариев, когда требуются такие эффекты, как отскоки, остановка, пауза, перемотка назад или замедление
2	Анимация не контролируемая: только запуск и пауза	Полный контроль над анимацией (возможно использование таких фреймворков как Velocity, GSAP и т.д.)
3	Синхронизируется с частотой обновления экрана	Синхронизируется с частотой обновления экрана
4	Нет возможности синхронизировать несколько элементов на странице друг с другом	Забивается основной поток

Как видно из таблицы, использование технологий CSS и JavaScript дало больше возможностей для создания плавной контролируемой анимации, что привело к повсеместному применению анимации. На сегодняшний день анимация в веб-сайтах выполняет ряд основных функций.

Во-первых, облегчает восприятие некоторых процессов и структур в интерфейсе. Благодаря анимации смягчается резкий переход между разными состояниями интерфейса с помощью свойств анимированного объекта, длительности анимации и скорости.

Во-вторых, привлекает внимание. С появлением веб-анимации дизайнеры пользовались колоссальным потенциалом в плане привлечения внимания для создания, среди прочего, рекламных баннеров, что в некоторых случаях было излишним. В настоящее время анимации используются на сайтах СМИ, в том числе для создания визуального интереса, динамики: например, иллюстрации в материале могут изображаться прямо на экране или вводиться при помощи привлекательного для взгляда перехода.

В-третьих, анимация создает визуальное соответствие между действием пользователя и событиями в интерфейсе. Анимация «оживляет» интерфейс согласно собственным внутренним законам или подчиняясь законам физики, что делает его более доступным для понимания пользователями [2].

Однако анимация должна оставаться не основным инструментом веб-дизайна, отсутствие анимации не должно сказываться на работоспособности интерфейса или считываемости контента сайта.

Таким образом, веб-анимация меняет наше представление о современном сайте, открывая новые возможности для улучшения функциональности интерфейса. Изучение современного состояния веб-анимации, возможностей, является важным вопросом в становление веб-разработчиков и веб-дизайнеров [3–5].

Литература

1. Анимация в веб-дизайне: зачем и когда ее нужно использовать [Электронный ресурс]. – Режим доступа: <https://spark.ru/startup/webest/blog/12176/animatsiya-v-veb-dizajne-zachem-i-kogda-eyo-nuzhno-ispolzovat>, своб.
2. Беляев А.А. Анимация в дизайне интерфейса информационных сайтов // Медиаскоп. – 2014. – № 4. – С. 11.
3. Сидерхолм Д. CSS3 для веб-дизайнеров. – Litres, 2016. – 7112 р.
4. Непочатов В.Ю., Чеботарёва И.Б. Несколько способов решить проблему анимации с помощью SVG и CSS [Электронный ресурс]. – Режим доступа: http://openarchive.nure.ua/bitstream/document/1625/1/ShS_PMW-2016_33.pdf, своб.
5. Сравнение анимации средствами CSS и JavaScript [Электронный ресурс]. – Режим доступа: <https://developers.google.com/web/fundamentals/design-and-ui/animations/css-vs-javascript?hl=ru>, своб.



Никифорова Анжелика Игоревна

Год рождения: 1992

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4108

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: nikiforova.anz@gmail.com



Погорелов Виктор Иванович

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, д.т.н., профессор
e-mail: pogvic@mail.ru

УДК 67

ОБЗОР ИСПОЛЬЗОВАНИЯ ТЕХНОЛОГИЙ 3D-ПЕЧАТИ ДЛЯ СОЗДАНИЯ ОДЕЖДЫ

Никифорова А.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 рассмотрены методы создания одежды посредством 3D-печати, их технологии и практическое применение.

Ключевые слова: 3D-печать, одежда, технологии, применение.

На данный момент 3D-печать является одной из самых перспективных инноваций. До некоторых пор этот способ печати был больше востребован в промышленных и научных кругах, но сейчас трехмерная печать начала активно внедряться и в повседневную жизнь людей. Создание одежды, напечатанной на 3D-принтере, активно применяется в индустрии моды, кроме того нынешняя доступность технологий трехмерной печати для домашнего использования делает возможным печатать одежду самостоятельно. **Целью работы** являлось изучение текущего состояния технологий 3D-печати, используемых для создания одежды, и их практическое применение.

Технологии 3D-печати подразделяются на четыре базовых типа:

1. ламинирование – вырезание слоев материала с последующим их склеиванием;
2. гранулирование – склеивание или спекание частиц материала;
3. экструдирование – выдавливание расплавленного материала;
4. фотополимеризация – отверждение полимера ультрафиолетовым или лазерным излучением [1].

В таблице приведены существующие технологии 3D-печати каждого типа.

Таблица. Технологии 3D-печати

Тип печати	Технология печати
Ламинирование	LOM
Гранулирование	SLS, SLM, DMLS, EBM, SHS, CDP
Экструдирование	FDM, DOT Jet, SFF
Фотополимеризация	SLA, DLP, MDM, PolyJet

Среди них на данный момент наиболее распространены три технологии: FDM (моделирование посредством наплавления), SLA (стереолитография), SLS (селективное лазерное спекание).

Наибольшую популярность в создании одежды имеет технология SLS. В 2010 году нидерландский модельер Айрис ван Херпен создала свою первую коллекцию одежды, напечатанную по этой технологии. В последующих ее коллекциях регулярно присутствуют напечатанные модели одежды. Специально для этих коллекций был разработан материал TPU 92A-1 – термопластичный полиуретан, обладающий упругостью, прозрачностью, и устойчивостью к истиранию [2].

В 2014 году компанией Nervous System было создано платье Kinematics Dress по технологии SLS из нетоксичного нейлонового пластика. На изготовление платья было затрачено около 44 ч. Материал включил в себя 2279 пластиковых треугольников, скрепленных 3316 петлями-замками. Благодаря особому проектированию платье практически не нуждалось в постобработке.

Платье, созданное для Диты фон Тиз в 2013 году также по технологии SLS из нейлона, состоит из 17 гибких частей с тремя тысячами подвижных соединений. Элементы были соединены вручную, отшлифованы, окрашены черным красителем, а затем инкрустированы 13 тысячами черных кристаллов Svarovsky. На 3D-печать платья потребовалось около 400 ч, а его сборка заняла несколько сотен часов.

В коллекции Айрис ван Херпен 2016 года присутствует платье, изготовленное по технологии SLA. Изделие отпечатано на принтере ProX 950 из прозрачного материала Accura ClearVue, и состоит из 6556 различных компонентов. Печать этого платья заняла около 200 ч.

Печать одежды по технологии FDM пользуется наименьшей популярностью, из-за низкого разрешения печати, и необходимости создания подпорок для печатаемых деталей. Однако это самый дешевый вид 3D-печати, что делает его доступным для домашнего использования. Печать одежды по технологии FDM была осуществлена дизайнером Данитой Фалек. Для печати использовался принтер Witbox, в качестве материала для печати использовалась нить FilaFlex, имеющая гибкие свойства. Предметы одежды имели структуру сетки [3].

Помимо использования традиционных технологий 3D-печати для создания одежды, идут разработки специализированных 3D-принтеров. Одной из таких разработок является проект Electroloom. Особенностью этого 3D-принтера является возможность печатать одежду, имеющую волокнистую структуру материала, схожего по фактуре с тканью. Материалом является смесь полиэфира и хлопка. Проект успешно завершил краудфандинговую кампанию, и в 2015 году был представлен прототип принтера, а также несколько напечатанных образцов одежды. Однако в 2016 году проект был закрыт, по словам автора, в связи с недостаточным финансированием [4].

Кроме того, существуют так называемые трикотажные 3D-принтеры, например, Kniterate. Он представляет собой вязальную машину, которая создает спроектированные на компьютере предметы одежды. Данная машина находится в настоящий момент в стадии готового прототипа.

Модельер Катя Леонович в 2015 году представила коллекцию, в которой большая часть одежды была из ткани, напечатанной на 3D-принтере. Печать ткани производилась по двум технологиям – SLS и FDM. С помощью FDM были напечатаны ткани, состоящие из множества соединенных колец, что напоминало кольчугу. При помощи SLS создавалось полотно, состоящее из переплетающихся нитей, что напоминает структуру ткани [5].

Особым направлением в развитии технологий создания одежды стоит выделить создание «умной» одежды, содержащей в себе устройства, позволяющие взаимодействовать с окружающей средой. В качестве примера можно привести так называемое «паучье платье», созданное инженером и дизайнером Анук Виппрехт. Данное платье, напечатанное по технологии SLS, имеет роботизированные конечности, которые способны менять положение в зависимости от показателей встроенных в платье датчиков приближения и дыхания. По задумке автора это платье создано, чтобы оградить личное пространство владелицы [6].

Дизайнеры Педро Оливейра и Ксуэди Чен создали платье, умеющее менять прозрачность. Оно состоит из гибкого напечатанного каркаса в виде сетки, и реактивных дисплеев. В зависимости от активности хозяйки в социальных сетях, телефон передает сигнал через Bluetooth платье, и тот или иной участок платья становится прозрачным.

Таким образом, 3D-печать постепенно внедряется в нашу жизнь, и имеет огромные перспективы развития для создания новых, более разнообразных или совершенных видов одежды.

Литература

1. Лысыч М.Н., Шабанов М.Л., Качурин А.А. Обзор современных технологий 3d печати // Современные наукоемкие технологии. – 2015. – № 6. – С. 26–30.
2. Рукавишников А.С., Евсева А.А. Инновационные технологии 3d-печати в сфере индустрии моды // Научно-методический электронный журнал концепт. – 2015. – Т. 13. – С. 3301–3305.

3. Are You Ready to 3D Print Your Own Clothes? [Электронный ресурс]. – Режим доступа: <https://3dprintingindustry.com/news/are-you-ready-to-3d-print-your-own-clothes-54620/>, своб.
4. Повод Г.В., Куренова С.В. Анализ и возможность применения 3D принтера в текстильном производстве // Инновационное развитие легкой и текстильной промышленности. – 2015. – С. 33–36.
5. Katya Leonovich Debut's 3D Printed Clothing Line at New York Fashion Week [Электронный ресурс]. – Режим доступа: <http://3dprintingindustry.com/news/katya-leonovich-debuts-3d-printed-clothing-line-new-york-fashion-week-32739/>, своб.
6. The Worst Date You Can Imagine Just Got Fully Horrible – The 3D Printed Attack Dress [Электронный ресурс]. – Режим доступа: <https://3dprint.com/32775/3d-printed-attack-dress/>, своб.



Нифанина Анастасия Романовна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: nanssea@yandex.ru



Шуклин Дмитрий Анатольевич

Год рождения: 1979

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна,

к.педагог.н., доцент

e-mail: do@limtu.ru

УДК 004.92

**ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ РАЗВЕРТОК ОБЪЕМНОЙ УПАКОВКИ
СУВЕНИРНОЙ ПРОДУКЦИИ**

Нифанина А.Р., 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 выделены этапы создания 2D-макетов 3D-упаковки, которые возможно в дальнейшем использовать в качестве схемы для написания технического задания для разработки макетов сувенирной продукции. В результате проведенного анализа графических приложений даны рекомендации по выбору графических редакторов для создания оригинал-макетов с учетом технологических возможностей полиграфической отрасли.

Ключевые слова: дизайн, упаковка, развертка, брендинг, проектирование, программное обеспечение, сувенирная продукция.

Упаковка – это средство или совокупность средств, которые защищают продукты от окружающего воздействия, дефектов и утраты в целом, облегчают процесс транспортировки,

хранения продукции, а также, благодаря яркому визуальному представлению, способствуют увеличению продаж.

Наполняемость может быть самой различной: от кондитерских изделий и региональных пищевых продуктов до продукции местных мастеров народных промыслов. Ведь основной задачей упаковки считается – реклама и повышение узнаваемости и лояльности покупателей к продукту на рынке. Красивая, качественная и запоминающаяся упаковка для любого товара – это как стильная, трендовая одежда «от кутюр». Привлекательно оформленный товар в интересной, яркой индивидуальной упаковке продается успешнее, интенсивнее и по более высокой цене, чем аналогичный товар в примитивной упаковке или без нее.

Использование продуманной упаковки, с ярким, достойным оформлением и необходимой рекламной информацией позволяет ускорить процесс продажи товаров, облегчить их потребление, а также оказывает сильное эстетическое воздействие на потребителей.

Интересным сегментом упаковочной продукции является сувенирная упаковка. Это может быть упаковка для сувениров с региональным компонентом в рамках брендинга территорий (тарелки с росписью, щепные птицы, чай, сушеные ягоды, кондитерские изделия). Идеальной упаковкой для подобного рода сувениров могут быть 3D-модели узнаваемых памятников архитектуры определенного региона (рисунок).



Рисунок. Архангельский гостиный двор: упаковка и памятник архитектуры

Задачи, выполняемые в рамках разработки упаковочного дизайна можно условно разделить на два взаимосвязанных направления: инженерно-функциональное формообразование и дизайнерское формообразование.

Инженерно-функциональное формообразование основывается на трех важных составляющих любой новой продукции – материале, конструкции и технологии производства [1].

В основе дизайнерского формообразования лежат теории композиции и эргономики.

Цель композиции в проектировании – создание оптимальных форм, которые отвечают требованиям производства и в экономическом плане, и по технологии, соответствуют функциональному назначению продукции, учитывают человеческий фактор и обладают эстетическими свойствами.

Художественное и инженерное моделирование взаимодополняют друг друга и представляют собой части единого процесса создания товарной упаковки. Все стадии художественного конструирования не стоит воспринимать как систему улучшения эстетики уже созданной упаковочной продукции. Процесс должен быть продуман и реализован на всех стадиях проектирования. Таким образом, графический дизайн упаковки и логотипа, разработанный благодаря сегодняшним компьютерным технологиям помогает реализовать задачу по комплексному созданию готового продукта.

Далее следует остановиться как раз на выборе программных технологий для инженерно-функционального и дизайнерского формообразования.

Изначально необходимо создать форму будущей развертки в специальном приложении. Специализированные CAD/CAM системы, такие как Impact CAD/CAM фирмы Arden

Software, ArtiosCAD фирмы ESKO, фирмы Elcede и другие, используют модули проектирования чертежа упаковки.

Среди универсальных систем автоматизированного проектирования (САПР) наибольшее распространение получила система AutoCAD компании Autodesk. Стоит отметить, что применение других программных средств (например, программ работы с векторной графикой) считается дилетантским путем, который лишен системности. Файл, полученный на этапе создания чертежа развертки упаковки, в дальнейшем станет основой файлов, которые управляют работой автоматизированного оборудования, поэтому качество выполнения чертежа считается основополагающим моментом. Для завершения разработки чертежа, специализированные САПР используют специальные функции (например, функцию удаления «двойных» линий), поэтому даже в случае использования универсальных САПР для разработки новой конструкции предпочтительным является последующий «прогон» файла через «сито» специальных функций модуля разработки чертежа развертки упаковки в составе специализированной CAD/CAM-системы. Это предъявляет к специализированным программным средствам необходимость импорта данных в формате универсальных САПР (в случае AutoCAD – файлы формата DXF) [2].

Далее разрабатывается непосредственно сам дизайн будущей упаковки. Популярные дизайнерские программы Adobe Photoshop, Adobe Illustrator, CorelDraw и так далее широко используются, но не являются специализированными для упаковки. Общие достоинства этих универсальных приложений – понятный интерфейс и богатейший инструментарий, позволяющий создать на экране монитора практически любой дизайн. Недостаток их в упаковочном производстве – низкая технологичность, т.е. отсутствие функций, которые учитывают специфику производства. Объединить эти приложения в технологическую линию непросто в силу их технических особенностей. Верстка – совмещение графического дизайна и чертежа развертки упаковки и внесение необходимых изменений в дизайн. Оптимальными для верстки являются специализированные CAD/CAM системы. Специализированные САПР содержат «облегченную» версию универсальных графических пакетов, которых вполне достаточно для этого этапа, так как графический дизайн уже не претерпевает здесь значительных изменений. Значит, основное требование к специализированным САПР – обеспечение возможности импортирования графики наиболее распространенных форматов.

Опытные образцы упаковки, как правило, производят на плоттере, оборудованном инструментальными головками для резки и биговки. При этом для управления современными плоттерами (плоттеры компаний Zund, Lasercomb) используется набор команд языка HPGL, поэтому, актуальной становится задача конвертации данных из формата САПР в формат HPGL. Большинство моделей плоттеров, не просто специализирующиеся для применения в индустрии упаковки из картона, а именно предназначенные для изготовления опытного образца упаковки (например, модель Lasercomb HSP), поддерживают возможность непосредственного управления из САПР, а также возможность работы в сети. В данном случае плоттер можно рассматривать как обычное сетевое устройство, и работа с ним подобна работе с сетевым принтером. Среди последних тенденций в изготовлении опытного образца упаковки из картона можно отметить применение лазерных технологий и технологий фрезерования, при этом вполне уместна работа как отдельных лазерной или фрезерной установок, так и модулей лазерной или фрезерной резки в составе плоттерной системы.

Мировым лидером в производстве допечатного обеспечения для упаковочной промышленности является компания ESKO, предлагая два редактора для подготовки упаковки к производству – ArtPro и PackEdge. В редакторах есть инструменты для работы с векторной и растровой графикой, средства верстки, набор специальных функций для учета технологических особенностей производства упаковки, и дополнительные модули, увеличивающие потенциал программного обеспечения. Предполагается, что они будут использоваться при окончательной компоновке дизайна – сборке различных элементов (текстовых, векторных, растровых) и выполнении технологических операций. Программы

самодостаточны, можно воспользоваться лишь одной на выбор для создания дизайна, поскольку включают важные функции для создания каждой составляющей упаковки, хотя уступают в удобстве интерфейса популярным универсальным приложениям.

Программы PackEdge и ArtPro предназначены для макетирования той упаковки, которая впоследствии будет печататься большими тиражами. Для макетирования упаковки сувенирной приемлемо использование классических графических пакетов. Для увеличения точности расчетов при создании разверток рекомендуется в дополнение к классическим использовать новую программу CorelCad, которая сочетает в себе расчетные и чертежные возможности AutoCad и Компаса и корректно работает с файлами расширения .CDR, что позволяет производить дизайн упаковки и макетирование ее развертки в одной программе [3].

Литература

1. Ефремов Н.Ф., Гротов А.С. и др. Автоматизированное проектирование упаковки. Учебное пособие. – М.: Университетская книга, Логос, 2008. – 248 с.
2. Cad в помощь [Электронный ресурс]. – Режим доступа: http://www.publish.ru/articles/201304_20013013, своб.
3. Дизайн упаковки: искусство или технология? [Электронный ресурс]. – Режим доступа: http://www.publish.ru/articles/200205_4045994, своб.



Павлихин Сергей Валерьевич

Год рождения: 1995

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: sergey.pavlikhin@inbox.ru



Готская Ирина Борисовна

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна,

д.педагог.н., профессор

e-mail: iringot@mail.ru

УДК 004

ИСПОЛЬЗОВАНИЕ UML ДЛЯ ПРОЕКТИРОВАНИЯ JAVASCRIPT-ПРИЛОЖЕНИЙ

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 преимущества JavaScript и UML, и их особенности. На основе этих данных сформулированы рекомендации по использованию UML для проектирования JavaScript-приложений.

Ключевые слова: UML, JavaScript, проектирование.

Язык JavaScript, который в основном применялся для разработки клиентской части клиент-серверных приложений (front-end), все больше заменяет такие популярные и общепризнанные

языки для разработки серверной части, как PHP, но и языки, которые совсем недавно стали применяться для программирования сервера, например, Java.

Так, согласно данным ежегодного отчета за 2016 год популярного форума Stackoverflow.com, 85,3% full-stack разработчиков, 90,5% front-end разработчиков и 50,4% back-end разработчиков предпочитают использовать язык JavaScript. Однако в 2015 году язык JavaScript был тоже самым популярным, процент использования был ниже – 54,4% [1]. Динамика роста популярности очевидна. Но с чем связана такая популярность?

Во-первых, язык JavaScript не требует компиляции. Если код исполняется на стороне клиента, то он просто загружается подобно HTML или CSS и сразу начинает свое выполнение. Это позволяет освободить серверную часть от дополнительной нагрузки. Эта же особенность позволяет делать очень интерактивные приложения. Во-вторых, этот язык прост для изучения. В нем нет сложных конструкций, и пишется на английском языке. В-третьих, все браузеры по умолчанию поддерживают этот язык. Однако не все браузеры одинаково обрабатывают один и тот же, и потому это требует от разработчиков писать кросс-браузерный код. Четвертое преимущество связано с тем, что JavaScript позволяет разрабатывать и клиентскую и серверную часть. Другие языки, которые используются в веб-разработке, обычно предназначены либо для серверной части, либо для клиентской. Например, PHP или Python работает только на стороне сервера, а ActionScript используется для программирования клиента [2].

К тому же, JavaScript продолжает развиваться и улучшаться. Если быть точным, то развиваются спецификации языка ECMAScript, на котором основан JavaScript. Последнее обновление было обнародовано в июне 2016 года [3]. Эти улучшения положительно сказываются на производительности языка, скорости разработки и приносят синтаксический сахар.

Развитие языка, рост популярности, а также сфер применения языка JavaScript делают очевидным проблему формального описания с целью последующей верификации и валидации в виде формальных моделей-приложений, которые были написаны на языке JavaScript. Язык JavaScript давно применяется разными разработчиками по всему миру. И потому, на данный момент существуют специфические паттерны, которые учитывают особенности языка JavaScript. Для проектирования нет специфичных инструментов, однако широко используется универсальный язык моделирования – UML (Unified Modelling Language).

UML позволяет разработчикам программного обеспечения достигнуть соглашения в графических обозначениях для представления общих понятий (таких как класс, компонент, обобщение (англ. «generalization»), агрегация (англ. «aggregation») и поведение) и больше сконцентрироваться на проектировании и архитектуре.

Сегодня он применяется для представления и проектирования систем, написанных на любых языках. UML считается стандартным языком моделирования, и многие компании требуют от разработчиков знание его.

Особенности языка JavaScript. JavaScript – объектно-ориентированный язык, однако его возможности объектно-ориентированности заметно отличаются от возможностей в C++ или Java, что некоторые даже считают его функциональным языком программирования.

Это связано в первую очередь с его особенностями, такими как, область видимости, функции, замыкания.

В JavaScript область видимости переменных задается не фигурными скобками, а определяется функциями, т.е., например, «if» и «for» не создают новую область видимости, а все объявления переменных в их конструкциях, на самом деле, «поднимаются». Это механизм в JavaScript называется «подъемом» (англ. «Hoisting» – поднимать), т.е. создается она в начале самой первой функции, в которой она объявлена, иными словами – в глобальной области видимости.

Функции в языке JavaScript являются объектами и могут использоваться разными способами. Например, функции могут присваиваться переменным и передаваться другими функциям. Поскольку функции являются объектами, имеется возможность присваивать значения их свойствам и даже вызывать их методы.

В JavaScript имеется полный доступ к внешним переменным и функциям внешней области видимости. Их можно считывать, записывать и, при необходимости, даже скрывать с помощью локальных определений. Созданная в замыкании функция «помнит» окружение, в котором она была создана. Комбинируя замыкания и вложенность функций, можно сделать так, что внешние функции будут возвращать внутренние без их исполнения.

Использование UML. Основная идея UML – возможность моделировать программное обеспечение и другие системы как наборы взаимодействующих объектов. Это, конечно же, замечательно подходит для объектно-ориентированных (ОО) программных систем и языков программирования, но также очень хорошо работает и для бизнес-процессов и других прикладных задач.

Но, так как JavaScript имеет не очень богатый набор возможностей ОО, то построение диаграммы классов необходимо делать с некоторыми договоренностями и соглашениями. Например, как известно, механизм наследования в JavaScript эмулируется с помощью прототипов. Для того чтобы это отразить, то можно показать зависимость между атрибутом конструктора «prototype» и объектом прототипом (рис. 1). Но можно и с помощью обобщения, если иметь об этом договоренность.

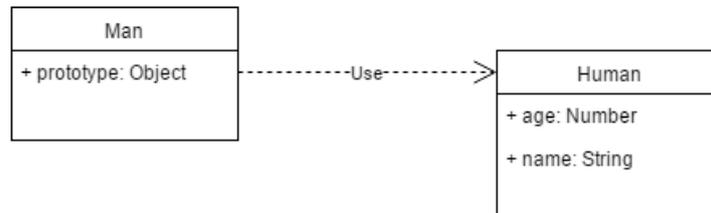


Рис. 1. Использование зависимости между prototype и объектом прототипом

В JavaScript часто функции возвращают другие функции, и все функции являются объектами типа «Function». И более того, JavaScript часто используют как набор последовательно вызываемых функций (функциональный стиль программирования), и потому возникают некоторые сложности при моделировании. И в этом случае необходимо проектирование специального профиля UML. Часть такого профиля представлена в таблице, а пример использования на рис. 2.

Таблица. Часть профиля UML для JavaScript

Элемент метамодели	Стереотип	Базовый элемент UML
function	«Function»	Class
function	«Call»	Association



Рис. 2. Пример построение диаграммы по профилю

В результате, независимо от того, в каком стиле написано приложение, UML способен описать архитектуру. Однако следует помнить, что при написании в функциональном стиле, возрастает время проектирования, потому что необходимо согласовать профиль между участниками разработки.

Несмотря на то, что ОО возможности JavaScript намного слабее, чем у других языков, UML все равно способен описывать архитектуру JavaScript-приложений. Причиной этому является гибкость и обширность UML, которая была заложена в основании языка [4–6].

Литература

1. Stack Overflow [Электронный ресурс]. – Режим доступа: <http://stackoverflow.com/research/developer-survey-2016>, своб.
2. Ecma International [Электронный ресурс]. – Режим доступа: <https://www.ecma-international.org/ecma-262/7.0/>, своб.
3. Гурьянов В.И. Профиль UML для имитационного моделирования [Электронный ресурс]. – Режим доступа: <http://cyberleninka.ru/article/n/profil-uml-dlya-imitatsionnogo-modelirovaniya>, своб.
4. Панкратова Т. Flash MX 2004: учебный курс. – СПб.: Питер, 2004. – 478 с.
5. Гвоздева Т.В., Баллод Б.А. Проектирование информационных систем. – Ростов-на-Дону: Феникс, 2009. – 508 с.
6. Рогозов Ю.И. Системный подход к созданию метода разработки информационных объектов на основе метамоделей // Информатизация и связь. – 2011. – № 7. – С. 57–62.



Панкратов Александр Сергеевич

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4206

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: sanyaone2033@yandex.ru

УДК 004.921

РАЗРАБОТКА МЕТОДА ПОВЫШЕНИЯ ВИДИМОСТИ САЙТА В ПОИСКОВЫХ СИСТЕМАХ

Панкратов А.С.

Научный руководитель – д.т.н., профессор Погорелов В.И.

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

Проведен анализ методов продвижения сайтов для внутренней оптимизации. Рассмотрены дополнительные методы продвижения, если стандартные методы не помогли продвинуть страницу в поисковых системах. На основе известных методов предложен свой метод для улучшения позиций сайтов.

Ключевые слова: SEO, WEB, продвижение, оптимизация, поисковые системы.

Внутренняя оптимизация является наиважнейшей частью при продвижении сайтов в поисковых системах. На данный момент основная сложность продвижения – это конкуренция по тематикам, к примеру, если сайт продает только индийскую одежду, то соответственно такой интернет-магазин легче встанет в первые позиции поисковых систем, но если тематика будет связана с продажей окон или по продаже автомобильных запчастей, то продвинуть страницы в поисковых системах будет сложнее.

Целью решения задачи явилось описание и разработка методов, которые позволят дополнительно помочь в продвижении сайтов в поисковых системах.

Проблема состоит именно в том, что для сайтов, которые имеют большую конкуренцию, как правило, недостаточно проведения стандартных мероприятий по их

продвижению. В данной работе описаны уже существующие методы, а также разработаны на основе существующих методы для эффективного продвижения сайтов.

Если углубиться в вопрос, то будут решаться проблемы внутренней оптимизации. Основные методы для продвижения страниц можно разделить на следующую структуру:

1. составление сематического ядра;
2. подготовка задания копирайтеру;
3. написание статей по теме;
4. подготовка текстов для размещения их на сайте;
5. размещение основных текстов на сайте;
6. сбор позиций страниц спустя какое-то время (это зависит от обновления поисковых систем, к примеру, на момент выполнения данной работы Яндекс обновлялся 2 раза в неделю).

Это стандартный комплекс мер для продвижения сайтов, если данные способы не помогают в выведении страниц в топ-позиции, то проводится следующие манипуляции:

1. товарная перелинковка;
2. изменение meta-тегов;
3. добавление тематических картинок с прописываем у них параметров alt и title по ключу.

Это не полный комплекс мер по продвижению. Например, здесь не учитывается анализ сайтов-конкурентов, но в рамках данной работы этого будет достаточно.

Предлагаемый метод предполагается использовать именно тогда, когда основные действия были сделаны для продвижения. Суть метода заключается в добавлении в текст, который уже был проиндексирован поисковой системой, выражений и дополнительных слов для увеличения «веса» страницы.

Выражения и дополнительные слова собираются на основе анализа конкурентов, так как при грамотном внедрении они положительно воспринимаются поисковыми системами. Это положительно скажется на продвижении тех ключей, которые имеют плохие позиции после основных манипуляций.

Рассмотрим подробнее методы внутренней оптимизации.

1. Составление семантического ядра. Для составления ядра семантики используется программа Key Collector.

Key Collector – это программа, которая сильно помогает при сборе ключевых фраз для составления семантического ядра. Приложение очень точно определяет конкурентность запроса и предположительную «дороговизну» его продвижения. В ходе анализа программа оценивает более 70 различных параметров и использует 4 метода взаимодействия с Wordstat, в отличие от других неэффективных решений, которые работают с готовыми базами данных сомнительной актуальности, Key Collector собирает информацию в режиме реального времени с сервисов-источников [1].

Wordstat – это онлайн-сервис, предоставляющий данные о том, сколько раз интернет-пользователи вводили в поисковую строку тот или иной запрос за последний месяц [2].

2. Задание копирайтеру. Основной и главный момент во внутренней оптимизации – является написание текстов. От них зависит дальнейшее продвижение страниц сайта в поисковых системах.

Задание составляется по следующим правилам:

- составляется список ключей на продвигаемой странице;
 - изменение ключей под грамотный язык;
 - уникальность текста больше 90% [3];
 - добавление слов, которые не вошли в группу ключей. Это не обязательный пункт.
3. Написание статей по теме. SEO-статья – это такая статья, которая написана не только для людей, но и для поисковых систем, т.е. – это статьи, с помощью которых

поисковые системы определяют тему в данной статье, чтобы в дальнейшем, при индексации страницы и выдаче ее пользователю, который ввел ключевой запрос, имеющий прямое отношение к данной статье, выдать ему нужную информацию, которая по мнению поисковиков присутствует на сайте [4].

Для страницы <http://lk-deti.ru/allcatalog/boy> были созданы следующие статьи:

- Как правильно выбрать одежду для мальчика?
- Какая подойдет детская одежда для мальчиков?
- На что стоит обратить внимание при выборе модной одежды для мальчиков?

Жирным шрифтом выделены ключи, на которые будут писаться статьи копирайтером.

4. Подготовка текстов. Данный этап заключается в подготовки текстов для сайта, написанных копирайтерами. Основные этапы подготовки текстов:

- прописывание тегов заголовков (H1, H2 и т.д);
- прописывание маркированных или нумерованных списков;
- прописывание мета-тегов (Title, description, key);
- проставление тематических картинок для сайта, с прописыванием для них параметров alt и title.

5. Размещение основных текстов на сайте. Эту часть продвижения нет необходимости подробно рассматривать. Смысл заключается в том, что подготовленные тексты отправляются клиенту на размещение их на сайте, либо клиент может попросить фирму, которая продвигает его сайт, разместить данные тексты.

6. Сбор позиций. Такие сервисы, как allpositions.ru и topvisor.ru предоставляют возможность проверить позиции ключей на сайте. Это необходимо для того, чтобы можно было проанализировать текущую ситуацию по продвижению [5].

Далее рассмотрим методы, которые используются дополнительно, когда основные способы не помогли улучшить позиции сайта.

1. Товарная перелинковка. Этот вид перелинковки используется для изменения названий товаров. Названия товаров должны максимально соответствовать продвигаемому ключу.

2. Изменение meta-тегов. Данный этап необходим, если meta-теги были прописаны заказчиком до того, как было им решено обратиться в фирму по продвижению. Как правило, заказчик не знает, как грамотно они должны быть оформлены.

3. Добавление тематических картинок с прописываем у них параметров alt и title по ключу. Добавление картинок используется для наполнения текстов визуальной информацией, такой текст пользователю будет приятнее читать, соответственно на странице он дольше задержится, и это положительно влияет как поведенческий фактор на продвижение [6].

Все перечисленные методы в совокупности очень хорошо помогают в продвижении страниц на сайте, но очень часто бывают ситуации, когда и этого не бывает достаточно. Как правило, в таком случае приходится писать заново тексты для страниц и проделывать все пункты, начиная со второго, а именно, составления задания копирайтеру. Предлагаемый метод позволит избежать ситуации написания новых текстов, основан он на анализе конкурентов выражений и дополнительных слов.

Для начала необходимо выбрать несколько выражений и дополнительных слов. Отличаются они тем, что выражение, как правило, состоит из двух слов а дополнительное слово только из одного. Собираются они следующим образом:

- вводиться ключ в поисковой системе;
- у первых 10 сайтов смотрятся сниппеты. Сниппет (от английского snippet – отрывок, фрагмент) – небольшая текстовая информация, относящаяся к сайту, которая выводится в результатах поиска;

– ключ, который был введен в поисковую систему в сниппетах, выделяется жирным шрифтом, рядом с ключом как раз и находятся выражения и дополнительные слова, которые необходимо собрать.

Приведем пример. Введя в поисковую систему ключ «автозапчасти», можно увидеть в сниппете первого сайта следующую фразу: «Интернет-магазин автозапчастей и аксессуаров...». Жирным здесь выделен ключ, а фразы по бокам являются его выражениями и дополнительными словами.

Подобрав и выбрав список таких слов, их необходимо внедрить в текст. Поисковые системы чувствительно относятся к околключевым фразам, и в теории это положительно скажется на продвижении страницы в топ.

Для улучшения позиции будут произведены методы, описанные ниже:

1. добавление на страницу выражений и дополнительных слов, которые уникальны для данных ключей страницы;
2. добавление выражений и дополнительных слов в текст.

Определенных правил добавления выражений и дополнительных слов нет, но нужно их внедрять с осторожностью, не задевая другие ключи. Лучше всего начать с добавления их в заголовки, а потом уже внедрять аккуратно в текст.

Спустя некоторое время планируется снять позиции сайта и посмотреть, как они изменятся. Сравнить их с первоначальными данными и сделать промежуточный вывод. Промежуточными они являются потому, что по одному сайту нельзя сделать вывод, нужно какое-то количество сайтов с общей тематикой. Проанализировав уже их, можно будет сделать окончательные выводы, а именно, работает предлагаемый метод на практике или нет.

Литература

1. MyDiv [Электронный ресурс]. – Режим доступа: <http://soft.mydiv.net/win/download-Key-Collector.html>, своб.
2. SEMANTICA [Электронный ресурс]. – Режим доступа: <https://semantica.in/blog/wordstat-yandex.html>, своб.
3. Jerri L. Ledord. Search Engine Optimization 2nd Edition // The Content Piece and Puzzle. – 2015. – P. 227–228.
4. SEMANTICA [Электронный ресурс]. – Режим доступа: <https://semantica.in/blog/kak-napisat-seo-statyu-poiskovaya-optimizaciya-statej.html>, своб.
5. ALLPOSITIONS [Электронный ресурс]. – Режим доступа: <http://allpositions.ru>, своб.
6. Fortress-Design [Электронный ресурс]. – Режим доступа: <http://www.fortress-design.com/vliyanie-graficheskogo-kontenta-na-prodvizhenie/>, своб.



Певзнер Виталий Владимирович

Год рождения: 1976

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: vpevzner@mail.ru



Погорелов Виктор Иванович

Год рождения: 1940

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, д.т.н., профессор
e-mail: kpd@limtu.ru



Шуклина Алла Сергеевна

Год рождения: 1981

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4206
Направление подготовки: 09.04.02 – Информационные системы
и технологии
e-mail: alla_shuklina@mail.ru

УДК 004.588

**ВЫБОР ПЛАТФОРМЫ ДЛЯ РАЗРАБОТКИ ИНТЕРАКТИВНОГО УЧЕБНИКА
С ЭЛЕМЕНТАМИ ГЕЙМИФИКАЦИИ**

Певзнер В.В., Погорелов В.И., Шуклина А.С.

Научный руководитель – д.т.н., профессор Погорелов В.И.

Работа выполнена в рамках темы НИР № 615862 «Фундаментальные исследования процессов взаимодействия излучения с новыми функциональными, конструкционными и биологическими материалами».

В работе изучена проблема выбора платформы для построения геймифицированного учебника на базе веб-технологий. Указаны основные направления разработки учебника, трудности при их реализации. Выбран инструмент для разработки учебника с элементами геймификации.

Ключевые слова: геймификация, дистанционное обучение, интерактивный учебник, игрофикация, геймифицированный учебник.

При разработке учебников с элементами геймификации наряду с подбором методических материалов, выбором игровых элементов, описанием основных игровых положений (возможные положительные и отрицательные ситуации, мотиваторы и демотиваторы обучающегося, система наград и наказаний и др.) и написанием сценария важную роль играет выбор платформы, на которой он будет реализован.

Если базироваться на интернет-технологиях, а также учитывать указанные в [1] особенности применения геймификации, то возникают три очевидных направления разработки.

1. Самостоятельное написание программного кода, реализующего сценарий учебника.
2. Использование системы управления обучением (LMS, Learning management system).
3. Использование системы управления содержимым (CMS, Content management system).

Рассмотрим достоинства и недостатки каждого из способов. При этом будем предполагать наиболее распространенную ситуацию – учебный курс разрабатывается преподавателем, не специализирующемся на использовании веб-технологий.

– Самый непростой путь, позволяющий получить качественный продукт, но с большими затратами времени. Для оптимизации времени выполнения требуется программист. Минимальные знания (HTML, CSS) без использования навыков программирования позволяют сравнительно быстро разработать качественный учебник, но реализация игровых элементов будет невозможна.

- На первый взгляд – это наиболее удобный способ, адаптированный под образовательные задачи. Существует ряд платных и бесплатных платформ, предназначенных для поддержки учебного процесса (Moodle, Sakai, Blackboard и др.). Они позволяют создавать учебники различной структуры [2]. Однако при попытке реализовать игровые технологии может возникнуть ряд проблем. Прежде всего, учебник, разработанный в такой оболочке, будет иметь стандартное оформление, и реализовывать обычную логику перелистывания (в лучшем случае, зависящую от результатов промежуточного тестирования), реализация же игровых элементов будет затруднена. Кроме того, учебник должен быть интегрирован в учебный процесс, управляемый выбранной LMS. Иначе говоря, внедрение курса в другое учебное заведение, базирующееся на иной LMS равносильно написанию учебника заново. Это сильно ограничивает тиражирование и применимость курса.
- Наиболее универсальный путь, позволяющий реализовать любой алгоритм учебника. Существует много как коммерческих, так и бесплатных систем управления содержанием, предъявляющих самые разные требования к квалификации разработчика. Выделим среди них наиболее известные: WordPress, Drupal, Joomla. В статьях, посвященных сравнению CMS [3, 4], упоминаются различные преимущества тех или иных систем перед конкурентами. Если главными критериями выбрать простоту освоения, распространенность, наличие обучающих материалов и уровень знаний разработчика в области веб-технологий, то оптимальным инструментом для реализации поставленной задачи можно назвать CMS WordPress.

Литература

1. Певзнер В.В., Погорелов В.И., Шуклин Д.А. Некоторые особенности применения геймификации в процессе обучения [Электронный ресурс]. – Режим доступа: <http://www.pmedu.ru/images/Documents/ps016-2/98-101.pdf>, своб.
2. Григорьев-Голубев В.В., Васильева Н.В., Ипатова Л.П., Леора С.Н., Певзнер В.В. Элементы дистанционного обучения математическим дисциплинам в системе высшего образования для кораблестроительных специальностей и направлений // Морские интеллектуальные технологии. – 2014. – Т. 1. – № 4(26). – С. 136–142.
3. Бучельников А.А. Сравнительный обзор систем управления содержанием сайтов // Современные технологии поддержки принятия решений в экономике. Сб. трудов Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых. Национальный исследовательский Томский политехнический университет. – 2015. – С. 210–212.
4. Новиков Н.С., Эфенди Ф.И. Разработка сайтов на основе систем управления контентом (CMS) // Молодежный научно-технический вестник. Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет). – 2015. – № 10. – С. 30.



Передрий Ольга Сергеевна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: krakoryak@gmail.com



Перепелица Филипп Александрович

Год рождения: 1977

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, ст. преподаватель
e-mail: phiper15@yandex.ru

УДК 004.55

МЕТОДЫ РАЗРАБОТКИ НА ПЛАТФОРМЕ ANGULARJS V2

Передрий О.С., Перепелица Ф.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 современном мире, где разработка программного обеспечения развивается быстрыми темпами, наибольшую популярность приобретают веб-приложения. На настоящий момент наиболее популярным фреймворком для написания веб-приложений является AngularJS. 15 сентября 2016 года состоялся релиз второй версии данного фреймворка, содержащий в себе кардинальные отличия. В связи с этими изменениями поменялись подходы к разработке. В работе рассмотрены возможные методы использования фреймворка Angular 2 для разработки веб-приложений.

Ключевые слова. Angular 2, SPA (Single Page Application), Model-View-Controller (MVC), Model-View-View-Model (MVVM), Model-View-Presenter (MVP), Document Object Model (DOM), Representational State Transfer application programming interface (REST API), фреймворк, Spring.

Angular 2 – фреймворк от компании Google, который позволяет разрабатывать сложные SPA-приложения (Single Page Application). Написание приложения состоит из создания шаблонов с помощью языка разметки HTML, написания классов-компонентов для работы с данными шаблонами, а также выделения общей логики приложения в сервисы. Все вместе это можно объединять в отдельные модули. Angular 2 написан в соответствии с шаблоном проектирования «модуль», что позволяет изолировать части логики приложения от глобального контекста, и содержит собственную систему модулей (NgModules) с тем, чтобы приложение имело понятную структуру и позволяло переиспользовать необходимые части без циклических зависимостей. Angular 2 написан на языке программирования TypeScript, разработанным компанией Microsoft в 2012 г., и являющимся надмножеством над языком JavaScript. Angular 2 предоставляет возможность написания веб-приложения с помощью следующих языков программирования [1, 2]:

- TypeScript;
- Dart;
- JavaScript (ES5/ES6).

Данный фреймворк позволяет реализовывать не только веб-приложения, но также мобильные и десктопные приложения, что делает его кроссплатформенным.

Таким образом, основными компонентами Angular 2, увеличивающими уровень абстракции, являются:

- модули (Modules);
- компоненты (Components) – управляют отображением данных в приложении;
- шаблоны (Templates) – шаблоны отображения данных;
- метаданные (Metadata) – описания поведения компонентов и классов;
- привязка данных (Data binding) – удобный способ организации взаимодействия шаблонов и компонентов, представлен в трех видах one-time binding (предоставление данных из модели в

представление, без отслеживания изменений «на лету»), one-way binding (позволяющий отслеживать изменения в модели и отображать их в представлении), two-way binding (позволяющий отслеживать изменения как в модели и отображать их в представлении, так и наоборот);

- директивы (Directives) – компоненты для расширения возможностей шаблонизатора;
- сервисы (Services) – компоненты для обеспечения работы логики приложения;
- внедрение зависимостей (Dependency injection) – обеспечение экземпляры компонентов внешними зависимостями;
- роутер (Router) – компонент для обеспечения навигации по приложению;
- формы (Forms) – компонент для работы с вводимыми пользователем данными;
- анимация (Animations) – компоненты для анимации пользовательского интерфейса;

Более того, приложения, написанные с помощью Angular 2, легко тестировать не только с помощью специальной утилиты Angular Testing Utilities, создающей тестовое окружение для приложения, но и с помощью широко известных фреймворков для тестирования, таких как Jasmine, Karma и Protractor.

В настоящий момент наиболее распространенными стеками разработки сложных приложений являются связки Angular 2 + ASP.NET MVC и Angular 2 + Spring MVC. В связи с тем, что Angular 2 написан на языке TypeScript (Microsoft), разработка решений с ASP.NET MVC (Microsoft) получила широкое распространение.

Машинный интерфейс корпоративного приложения, обладающего клиентской частью, выгодно писать как REST API. Этот машинный интерфейс содержит обычные уровни:

- уровень маршрутизации: определяет, какие точки входа сервисов соответствуют конкретным HTTP URL, и как будут считываться параметры из HTTP-запроса;
- уровень сервисов: содержит лишь бизнес-логику (например, обеспечивает валидацию), определяет область применения бизнес-транзакций;
- уровень сохраняемости: отображает базу данных на объекты предметной области, хранящиеся в памяти и наоборот.

Такие связки позволяют по-новому взглянуть на разработку веб-приложений, связанных с интенсивным заполнением форм. Данный подход довольно хорошо повышает продуктивность за счет создания машинного интерфейса с REST API. Более того, этот вариант использования JavaScript в интерфейсной части и Java или .NET для работы с базой данных также повышает продуктивность работы приложения. Дополнительным плюсом является отсутствие привязки к состоянию сервера между запросами, что помогает избежать ряд ошибок.

Литература

1. Fain Y., Moiseev A. Angular 2 Development with TypeScript. – Manning Publications, 2016. – 456 р.
2. Angular 2 [Электронный ресурс]. – Режим доступа: <https://angular.io/>, своб.



Перепелица Филипп Александрович

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, ст. преподаватель
e-mail: phiper15@yandex.ru



Калимуллина Анастасия Юрьевна

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4206

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: globina93@gmail.com

УДК 004.92

УРОВНИ ДЕТАЛИЗАЦИИ ИНФОРМАЦИОННОЙ МОДЕЛИ НА РАЗЛИЧНЫХ ЭТАПАХ ЖИЗНЕННОГО ЦИКЛА

Перепелица Ф.А., Калимуллина А.Ю.

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

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

В настоящее время для организации процесса планирования BIM-проекта, осуществления 3D-координации и передачи необходимой информации для решения основных проектных задач активно используется концепция LOD. В данной работе проведен обзор концепции применения различных уровней детализации информационной модели на различных этапах жизненного цикла.

Ключевые слова: LOD, BIM, уровни детализации информационной модели.

LOD («Levels of Detail») – это прием, который используется в проектировании, заключающийся в создании вариантов одного объекта с различными степенями детализации в зависимости от этапа жизненного цикла изделия или строительного объекта, в частности. Активно используется в информационном моделировании зданий (BIM) [1].

Уровни детализации информационной модели определяют полноту проработки элемента информационной модели, они задают минимальный объем геометрической, пространственной, количественной, а также любой атрибутивной информации, необходимой и достаточной для решения задач моделирования.

Уровни детализации должны соответствовать каждому этапу проекта в зависимости от потребностей каждого участника проекта. Так, например, на этапе подготовки сметы нет необходимости детального описания каждой составляющей инженерного оборудования, для сметчика не имеет значение, как именно будет выглядеть определенная позиция по смете, какие габариты она будет иметь и так далее, но при этом важно указать производителя, маркировочный номер по каталогу и стоимость. Но в то же время для инженеров важно знать габаритные размеры и основные технические характеристики.

Следовательно, основной задачей применения LOD является получение минимальной, но в то же время достаточной информации для каждого участника информационного моделирования, что способствует решению определенных задач на каждой стадии проекта.

В 2004 году концепцию уровней детализации «Level of Detail» представила компания Vico Software, которая специализируется на производстве программного обеспечения для управления строительными проектами и выполнения сметных расчетов. Основное назначение LOD компания Vico Software в то время видела в определении стоимости строительства на различных стадиях проекта.

Изначальная трактовка термина «Level of Detail» определяла уровень детализации как степень графической и информационной насыщенности элементов модели, но не затрагивала вопрос минимальной достаточности этой информации для использования другими участниками проекта. Основной задачей было определение стоимости строительного проекта на различных

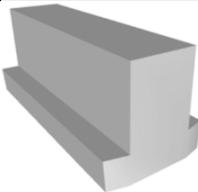
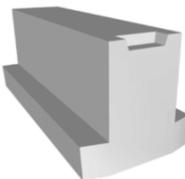
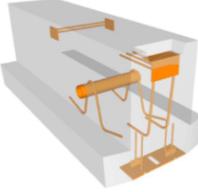
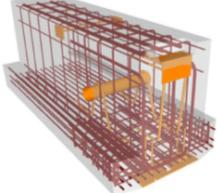
стадиях: от оценки приблизительной стоимости на концептуальной стадии до определения точной стоимости на стадиях выпуска рабочей документации и производства строительно-монтажных работ.

В 2008 году Американский Институт Архитектуры (AIA) развил концепцию LOD в виде протокола AIA E202-2008, Building Information Modeling Protocol Exhibit (типовая форма приложения к договору на BIM проект). На сегодняшний день действуют усовершенствованные версии этого протокола: AIA E203–2013 Building Information Modeling and Digital Data Exhibit и AIA Contract Document G202-2013 Building Information Modeling Protocol Form.

В этих документах уровень детализации был определен как «Level of Development» (уровень проработки) элементов модели. Это переименование термина «Level of Detail», было обоснованным, но, тем не менее, внесло некую путаницу в общую BIM-терминологию так как аббревиатура осталась такой же – LOD [2].

В настоящее время для организации процесса планирования BIM-проекта, осуществления 3D-координации и передачи необходимой информации для решения основных проектных задач было определено пять базовых уровней детализации элементов информационных моделей: LOD100, LOD200, LOD300, LOD400 и LOD500. Описание базовых уровней приведено в таблице.

Таблица. Описание LOD

LOD	Описание	Пример
LOD100	Концептуальный уровень, элементы модели могут быть вообще не представлены визуально, либо представлены в виде объемных формообразующих элементов с приблизительными размерами.	
LOD200	Элементы модели представлены визуально в виде объемных элементов с приблизительными размерами и пространственной ориентацией. Модель может содержать кроме геометрических параметров еще и минимальную атрибутивную информацию.	
LOD300	Элементы модели имеют точные геометрические размеры и пространственную ориентацию. Модель содержит необходимый объем атрибутивной информации.	
LOD400	Элемент модели представлен в виде сборки, принадлежащей конкретной системе с точными размерами, формой, пространственным положением и другой необходимой неграфической атрибутивной информацией.	
LOD500	Элемент модели представлен в виде сборки, принадлежащей конкретной системе с точными размерами, формой, пространственным положением и другой необходимой неграфической атрибутивной информацией, достаточной для передачи в эксплуатацию.	

Для успешного внедрения LOD в BIM-проект должны быть выполнены определенные условия.

LOD должны быть предварительно заданы при формировании документа «Информационные Требования Заказчика» (приложение к ТЗ на проектирование с применением

технологий BIM), исходя из основных целей использования информационных моделей. Цель – внести однозначность в понимание уровней детализации службами Заказчика и Проектной Командой.

LOD должны быть тщательно продуманы и задокументированы в виде «Спецификации Элементов Модели» при разработке документа «План Выполнения BIM-проекта». Цель – конкретизация проектных требований для членов проектной команды, а также планирование и управление рабочими BIM-процессами.

LOD должны обязательно быть «привязаны» к классификатору (системе кодирования).

LOD должны содержать минимальное, но достаточное для выполнения основных проектных задач, количество информации. Предоставление излишней информации в элементах моделей расточительно, и может серьезно затруднить работу с моделями и увеличить сроки проектирования [3].

Подготовка и выполнение BIM-проекта невозможны без предварительной детальной разработки BIM-стратегии компании, включающей пакет документов корпоративных стандартов, классификаторов, регламентов и процедур коллективной работы.

Литература

1. BIM-стандарт организации для площадных объектов (для Revit® и AutoCAD® Civil 3D®). Шаблон. – М.: Autodesk, Inc, 2015. – С. 109.
2. BuildingSMART [Электронный ресурс]. – Режим доступа: <http://buildingSMART.org/standards/activities/>, своб.
3. CONCURATOR [Электронный ресурс]. – Режим доступа: <http://concurator.ru/>, своб.



Проводников Андрей Дмитриевич

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4208
Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: andrewprovodnikov@gmail.com



Шуклин Дмитрий Анатольевич

Год рождения: 1979

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

e-mail: kpd@limtu.ru

УДК 004.057.2

ИНТЕРАКТИВНЫЕ ВОЗМОЖНОСТИ ВОСПРОИЗВЕДЕНИЯ МЕДИАКОНТЕНТА СРЕДСТВАМИ HTML5

Проводников А.Д.

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

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе рассмотрены проблемы воспроизведения медиаконтента в сети Интернет. В работе дан краткий обзор спецификации Media Source Extensions и ее возможностей.

Ключевые слова: HTML5, MPEG-DASH, MSE, медиаконтент, потоковое видео.

В настоящее время пользователи сети Интернет просматривают большое количество медиаконтента (обобщенное название видео- и аудиоматериалов) с различными целями на большом количестве различных устройств, что диктует определенные требования к методам работы с ней:

- кроссплатформенность и кроссбраузерность – воспроизведение медиаконтента в браузерах на различных устройствах;
- поддержка форматов – воспроизведение медиаконтента без необходимости изменения исходного формата;
- гибкость в управлении – перемотка медиаконтента без необходимости ожидания его полной загрузки, изменение битрейта медиаконтента;
- сведение медиаконтента – воспроизведение медиаконтента в необходимом порядке, вставка фрагментов одного медиаконтента в другое.

Исторически работа с медиаконтентом велась с помощью сторонних технологий, таких как Flash, Silverlight. Они позволяли реализовать разработчику и гибкость управления, и сведение материалов. Также они поддерживали большое количество форматов. При этом, являясь сторонними технологиями, Flash и подобные ему решения требовали установки (что также требовало поддержки данной технологии устройством клиента), а также являлись опасными с точки зрения безопасности.

Очевидно, что встроенные в HTML методы работы с медиаконтентом являлись бы решением данных проблем. В связи с этим в спецификацию HTML5 были добавлены медиаэлементы video и audio. Стандарт HTML5 стал официально рекомендуемым 28 октября 2014 года, а поддержка video и audio в браузерах Firefox, Safari была реализована уже в 2009 году. Плюсами элементов являются кроссплатформенность, кроссбраузерность, безопасность. Говоря о минусах, video и audio сильно ограничены в возможностях управления, как со стороны пользователя, так и со стороны разработчика.

Для решения этой проблемы была разработана технология адаптивной потоковой передачи данных MPEG-DASH [1]. Эта технология является первым решением по потоковой передаче данных с адаптивным битрейтом, которое получило статус международного стандарта [2]. В качестве поддержки новой технологии была создана спецификация HTML5, получившее название Media Source Extensions (MSE) [3]. 17 ноября 2016 MSE получило статус рекомендованного стандарта [4].

Спецификация разрабатывалась с учетом следующих целей:

- дать возможность конструировать медиа-потоки с помощью JavaScript в независимости от того, как происходит процесс получения источника;
- определить модель с учетом разделения медиа и буферизации для облегчения адаптивной передачи данных, вставки рекламы, перемотки медиаконтента и редактирования видео;
- эффективное использование кэша браузера;
- обеспечение требований к спецификации формата бинарных данных;
- отсутствие необходимости поддержки какого-то конкретного формата.

Media Source Extensions позволяет изменять разрешение медиаконтента в зависимости от действий пользователя, скорости передачи данных и т.п. Также в MSE предусмотрена возможность сводить фрагменты для конкретного пользователя, позволяя, например, показывать рекламу на основании данных об интересах пользователя. Благодаря MSE, возможно создание видеоконференций для различных устройств, где в качестве медиаконтента будет выступать видео и звук с устройства пользователя.

MPEG-DASH и MSE добавляют HTML5 интерактивные возможности для воспроизведения медиаконтента. Сравнивая с более старыми аналогами, такими, как Flash, возможности не выглядят кардинально новыми, но необходимо отметить, что только MPEG-DASH имеет статус международного стандарта, обладая также поддержкой на уровне HTML5, что делает новую технологию кроссплатформенной, кроссбраузерной и безопасной, не ограничивая ее в гибкости и возможностях.

Литература

1. Thang T.C. et al. Adaptive streaming of audiovisual content using MPEG DASH // IEEE Transactions on Consumer Electronics. – 2012. – V. 58. – № 1. – P. 78–85.
2. ISO/IEC 23009-1:2014 – Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats [Электронный ресурс]. – Режим доступа: http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=65274, своб.
3. Media Source Extensions™ [Электронный ресурс]. – Режим доступа: <https://www.w3.org/TR/media-source>, своб.
4. MEDIA SOURCE EXTENSIONS™ IS A W3C RECOMMENDATION [Электронный ресурс]. – Режим доступа: <https://www.w3.org/blog/news/archives/5979>, своб.



Пузырев Павел Константинович

Год рождения: 1992

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4206

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: puzyrev.pavel@yahoo.com

УДК 004.9

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АРХИТЕКТУРНЫХ ПАТТЕРНОВ В iOS РАЗРАБОТКЕ

Пузырев П.К.

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

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

В работе рассмотрены одни из самых распространенных архитектурных паттернов, которые применяются при проектировании и планировании архитектуры приложения. Проведен их анализ с раскрытием слабых и сильных сторон, приведен базовый список рекомендаций по применению каждого из шаблонов.

Ключевые слова: архитектурный паттерн, Сосоа, MVC, MVP, MVVM.

Для разработчика программного обеспечения – программиста – гарантом низкого порога вхождения в уже существующий проект является читабельный и разделенный на слои код. В связи с этим в дальнейшем представляется возможным переиспользование определенных частей кода в других проектах без каких-либо трудностей. Достичь такого результата возможно за счет грамотного подхода к описанию архитектуры программы или приложения на начальном этапе проекта, а именно, определиться с архитектурными паттернами, которые станут основой для разработки. Архитектурой

приложения является логическая структура, описывающая отдельные компоненты, их свойства и связи в виде единой системы. Паттерны – это описания схем детализации отдельных подсистем приложения и взаимосвязей между ними. При этом паттерны не являются частью программы, не влияют прямо на ее структуру и сохраняют полную независимость от языка программирования конкретной системы [1].

Основными характерными для iOS разработки маркерами отсутствия существования единого подхода к построению приложения являются: «божественный» класс ViewController размером в несколько тысяч строк, сохранение данных в самом ViewController, модель не является бизнес-логикой, отсутствие возможности проведения тестов.

Показателем хорошей архитектуры приложения является адекватное распределение обязанностей между сущностями, т.е. явно присутствует разделение на слои, возможность проведения тестирования, легкое внедрение дополнительных модулей и в дальнейшем простота использования существующего кода.

На сегодняшний день популярными являются следующие архитектурные паттерны:

1. MVC или Cocoa MVC (Model-View-Controller);
2. MVP (Model-View-Presenter);
3. MVVM (Model-View-ViewModel).

Для этих паттернов характерны три категории, в которые должны распределиться все классы проекта, причем только в одну:

1. Model (модель) – часть, которая содержит в себе бизнес-логику, просто хранит данные или, например, менеджер базы данных. Она не должна знать о View-контроллерах;
2. View (представление) – отображает данные, которые были получены от Model. Это все классы, которые начинаются с префикса UI;
3. Controller (контроллер), Presenter (презентер), ViewModel (модель представления) – обновляют View, изменяют model, обрабатывают события и т.д.

Доминирующим архитектурным паттерном среди начинающих разработчиков является MVC. Рассматривая его как парадигму, конечно, нельзя говорить о четкой его формулировке, ведь это, по сути, является лишь набором рекомендаций. «Apple» на официальном сайте рассказывает о своем видении MVC, а именно Cocoa MVC (рис. 1, а).

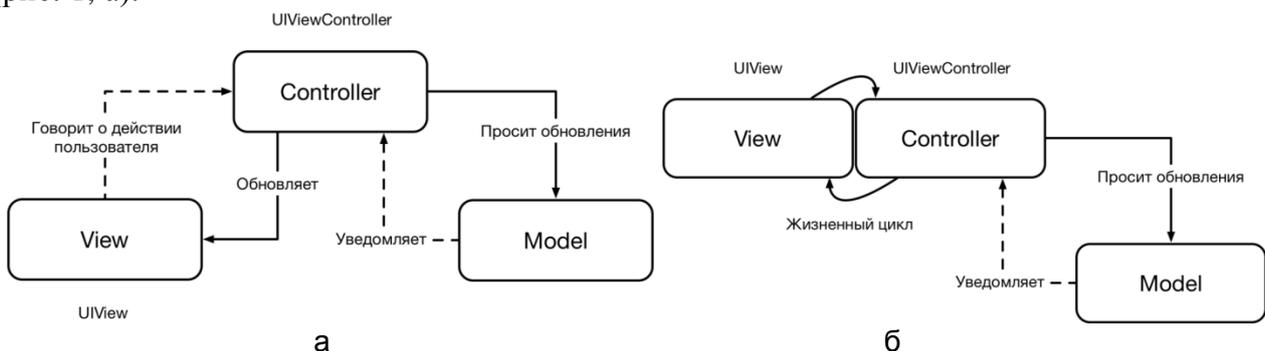


Рис. 1. Схематическое представление Cocoa MVC (а) и Cocoa MVC в действительности (б)

Как видно, View и Model никак не связаны между собой, а Controller является посредником между ними, что делает его труднопереиспользуемым. В свою очередь, это не является большой проблемой, так как, например, в нем можно разместить немного специфичной бизнес-логики.

На деле же все получается совершенно иначе из-за того, что Controller почти напрямую связан с View, поэтому считать их отдельными сущностями нельзя. Кроме

того, обычно Controller является источником данных, делегатом и многим другим, а View лишь отправляет события контроллеру (рис. 1, б).

Одним из паттернов, который предполагает разделение на три независимые категории является MVP. Именно в нем решается проблема Cocoa MVC, а под View понимается сама View или связка «View+View Controller» путем добавления презентера (рис. 2, а).

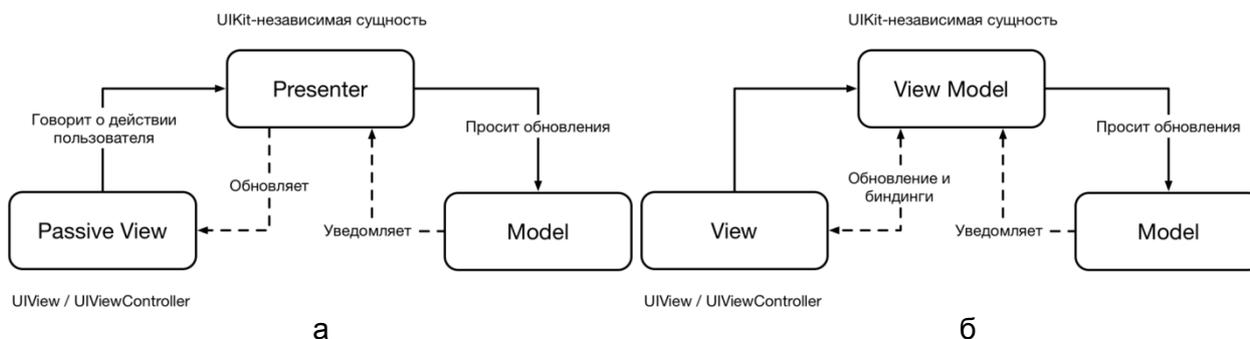


Рис. 2. Схематическое представление MVP (а) и MVVM (б)

Presenter – это сущность, которая не знает о UIKit и занимается только обновлением View вместо View Controller в Cocoa MVC, таким образом, разгружая его. Теперь у каждой View по мере необходимости будет свой Presenter для обновления, что удобно для восприятия кода проекта в целом.

Внешне похожим на MVP паттерном является MVVM. В нем также новая составляющая, ViewModel, не знает о UIKit, Model и View никак не связаны, а View Controller относится к View (рис. 2, б). Среди отличий от чистого MVP стоит отметить использование биндингов между ViewModel и View – это и есть главная особенность данного паттерна.

ViewModel является сущностью, которая вызывает обновление Model, например, в результате действий пользователя, после чего сама и обновляется с уже новой Model за счет биндингов. По умолчанию в iOS биндинги не реализованы, в качестве альтернативы разработчикам предлагается использовать KVO (Key Value Observing) и Notifications (уведомления), но это не является хорошей заменой. Среди сторонних библиотек для их реализации одной из самых популярных является «ReactiveCocoa», она же является фреймворком для реактивного программирования [2].

После детального рассмотрения этой тройки паттернов можно сделать следующие выводы:

1. Cocoa MVC является легким в реализации, поэтому его сможет поддержать разработчик любого уровня. Говоря про связь View и Controller, то тестированию подлежит лишь Model. Логично предположить, что его еще можно использовать в том случае, когда запрещается или не нет желания использовать биндинги;
1. MVP чуть сложнее в реализации, чем MVC, но обладает полностью независимой View, что позволяет проводить простую проверку бизнес-логики. Основная нагрузка ложится на Model и Presenter. Данный паттерн можно реализовать без биндингов, так и с ними – за счет дополнительной связи между View и Model. Но, несмотря на простоту идеи этого паттерна, априори количество кода увеличится в два раза, чем при MVC;
2. MVVM перенял все плюсы вышеупомянутых подходов. View обновляется сама с помощью биндингов, и ViewModel не знает о ней, что хорошо для тестирования. Сложность реализации может возникнуть, если в проекте запрещено использовать сторонние библиотеки, в таком случае придется пользоваться KVO и Notifications. По количеству кода данных подход будет почти идентичен MVP.

Следует также иметь в виду, что придерживаться строго одному подходу – далеко не лучшая идея, так как для реализации одной подзадачи, например, будет достаточно MVC, в то время как в проекте все организовано по MVVM. Целью планирования и выбора архитектуры приложения является легкость его будущей поддержки и тестирования [3].

Литература

1. Введение в архитектуру и паттерны программирования [Электронный ресурс]. – Режим доступа: <https://laravel.ru/posts/3/>, своб.
2. Паттерны ООП в примерах для iOS [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/company/badoo/blog/281162/>, своб.
3. iOS Architecture Patterns [Электронный ресурс]. – Режим доступа: <https://medium.com/ios-os-x-development/ios-architecture-patterns-ecba4c38de52/>, своб.



Савенкова Ирина Алексеевна

Год рождения: 1991

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107с

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: irina.savenkova@hotmail.com

УДК 004.558

ПОСТРОЕНИЕ ИГРОФИЦИРОВАННОГО ПРИЛОЖЕНИЯ НА ОСНОВЕ СТРУКТУРНОГО ПОДХОДА

Савенкова И.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 рассмотрена процедура построения игрофицированного приложения с позиции структурного подхода. Представлена схема соотношения игровых динамик, механик и компонентов. Особое внимание уделено модели PBL, наиболее часто используемой для геймификации приложений.

Ключевые слова: игрофикация, геймификация, игровые технологии, очки, бейджи, рейтинги, PBL, игровые динамики, игровые механики, игровые компоненты.

Актуальность работы обусловлена интересом профессионального сообщества к игрофикации приложений. Интерес, в свою очередь, вызван желанием создать востребованный информационный продукт, а именно использование игровых технологий в настоящее время позволяет привлечь и удержать пользователей приложения.

Важным аспектом игрофикации является достижение с ее помощью целей, напрямую не связанных с содержанием игры, например, отработка определенных навыков, вовлечение в выполнение рутинных дел, повышение производительности труда и другое. Актуальным является создание приложений для решения следующих задач: соблюдение диеты, занятие спортом, отказ от курения, поддержка корпоративной культуры, обучение. Таким образом, среди наиболее популярных объектов для игрофикации выступают спортивные, корпоративные и обучающие веб-приложения.

Игрофикация – это не отдельные игры и даже не совокупность игр, а общая игровая оболочка для какого-либо целенаправленного процесса. Результатом и целью внедрения такой игровой оболочки в жизнь является не просто повышение мотивации или интереса, а изменение системного поведения человека, группы людей, некой части или общества в целом. Большую роль в развитии игрофикации сыграли рост социальных сетей и распространение технических средств, связанных с быстрым обменом информацией: смартфонов, планшетов, нетбуков и т.д.

Существуют различные подходы к определению игрофикации. В самом широком смысле, геймификацию определяют как процесс применения игровых элементов и механизмов в неигровом контексте [1].

А.В. Цветчих, А.В. Редькина рассматривают игрофикацию как эффективный прием повышения мотивации индивидов [2].

А.Я. Казаков [3], Н.В. Аверина [3] и др. определяют «игрофикацию» как технологию, имеющую интерактивное и кроссмедийное наполнение, за счет которого возможен охват большого объема учебного материала и экономия времени в процессе обучения. Единая программная оболочка технологий игрофикации может создаваться с помощью chm-, flash-технологий, авторских программ, инструментальных средств общего назначения, языков программирования, технологий облачных вычислений и т.д.

Д.Ю. Санагурский определяет «игрофикацию» как сложный процесс, сочетающий в себе элементы психологических, культурологических и педагогических технологий для повышения мотивации пользователей игр [4].

М.А. Черепица определяет игрофикацию как концепцию, которая ориентирована на использование игровых механик вне контекста игры [5].

Структурный подход подразумевает расчленение системы на составляющие структурные элементы в их взаимодействии, поскольку в реальных условиях каждый структурный элемент воздействует как на все другие элементы, так и на систему в целом.

Наиболее популярной при игрофикации приложения является PBL-модель, включающая такие основные элементы, как очки, бейджи и рейтинги.

Очки выдаются за выполнение заданий, их можно обменять на вознаграждения. Очки могут использоваться для ведения счета, разграничения уровней, определения победы. К их достоинствам следует отнести демонстрацию прогресса пользователя, выполнение функции обратной связи.

Бейджи – это визуальное представление некоторых достижений в рамках геймифицированного процесса. Бейджи могут выдаваться за достижение пользователем определенного уровня или же определенный вид деятельности. Бейджи играют мотивирующую функцию, кроме того, они способствуют адаптации новых пользователей, поощряя определенное поведения, т.е. показывая, как следует пользоваться данной системой. Бейджи также показывают статус игрока и в определенных ситуациях формируют чувство общности.

Рейтинги позволяют отследить место пользователя в иерархии, таким образом, играют сильную мотивирующую роль при правильном использовании.

Согласно К. Вербаху [6] существуют три категории игровых элементов, применимых в геймификации: динамики, механики и компоненты. Элементы игры можно представить в виде иерархии. Внизу – компоненты, выше – механики игры, на вершине – динамики игры. Причем как каждая механика связана с одной или несколькими динамиками, так каждый компонент связан с одной или несколькими механиками.

Динамики – это своеобразные принципы построения игрофицированных систем. Это достаточно абстрактное понятие, которое формирует, прежде всего, подход к построению игрофицированной системы. Можно выделить следующие динамики: ограничения (лимиты или вынужденные компромиссы), эмоции (любопытность, дух соперничества), повествование (последовательная, непрерывная сюжетная линия), продвижение (рост игрока

и его развитие), отношения (социальные взаимодействия, формирующие чувства товарищества, статуса и альтруизма).

Механики – основные процессы, которые движут действиями и формируют у игрока вовлеченность. Можно выделить десять важных игровых механик: задания (загадки или любые другие задания, которые требуют усилий для их решения); шанс (элементы случайности); соревнование (один игрок или группа игроков побеждает, а вторая – проигрывает); сотрудничество (игроки должны работать вместе, чтобы достичь общую цель); обратная связь (информация об успехах игрока); накопление ресурсов (получение полезных или коллекционных предметов); вознаграждения (награды за определенные действия и достижения); сделки (торговые операции между игроками, напрямую или через посредников); ходы (поочередное участие меняющихся игроков); состояние победы (показатели, которые превращают игрока или команду в победителя; состояния выигрыша и проигрыша – связанные понятия).

Каждая механика – способ достижения одной или нескольких описанных динамик. Случайное событие, такое как неожиданное вознаграждение, может стимулировать в игроках вовлеченность и любознательность. Механики способствуют адаптации новых игроков, а также поддерживают интерес существующих игроков в соответствии с кривой интереса.

Компоненты – это более конкретная форма, которую принимают механики и динамики. Пятнадцать важных игровых компонентов: достижения (определенные цели); аватары (визуализация характера игрока); бейджи (визуализация достижений); битвы с боссами (особенно сложные испытания для перехода на следующий уровень); коллекционирование (накопление наборов предметов или бейджей); сражения (конкретная борьба, обычно быстрая); доступ к контенту (то, что открывается игрокам, когда они достигают определенных показателей); подарки (возможность делиться ресурсами с другими); рейтинги лидеров (визуализация развития и достижений игрока); уровни (определенные шаги в развитии игрока); очки (количественное отображение развития игры); квесты (конкретные задачи со своими целями и наградами); социальный профиль (визуализация игры в социальной сети игрока); команды (определенные группы игроков, работающих вместе ради общей цели); виртуальные товары (игровые активы с субъективной или реальной денежной ценностью).

Таким образом, создание игрофицированного приложения предусматривает использование трех категорий игровых элементов: динамик, механик и компонентов. Как каждая механика связана с одной или несколькими динамиками, так каждый компонент связан с одним или несколькими элементами высшего уровня. Наиболее простой метод внедрения игровых технологий в приложение – обращение к модели PBL, т.е. добавление в приложение очков, бейджей и рейтингов.

Литература

1. Маркеева А.В. Геймификация как инструмент управления персоналом современной организации // Российское предпринимательство. – 2015. – Т. 16. – № 12. – С. 1923–1936.
2. Цветчих А.В., Редькина А.В. Методы и приемы, повышающие качество подготовки студентов информационных специальностей в вузе // Решетневские чтения. – 2012. – № 2(16). – С. 647–648.
3. Казаков А.Я., Аверина Н.В., Дроздова Е.Н., Кайнарова Е.М. Разработка технологии создания интерактивных и кроссмедийных электронных учебных пособий // XIX Международная научно-методическая конференция «Современное образование: содержание, технологии, качество». – 2013. – С. 229–230.
4. Санагурский Д.Ю. Игрофикация (Gamification) как фактор формирования виртуальной идентичности [Электронный ресурс]. – Режим доступа: http://www.cr-journal.ru/rus/journals/251.html&j_id=18, своб.

5. Черепица М.А. Игрофикация. Игровые элементы в маркетинге будущего // Региональная научно-практическая конференция «Актуальные проблемы гуманитарных, социальных и экономических наук: вопросы теории и практики». – 2012. – С. 42–46.
6. Вербх К. Вовлекай и властвуй. Игровое мышление на службе бизнеса. – М.: Манн, Иванов и Фербер, 2015. – 208 с.



Сверчков Василий Александрович

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: vasilyars@gmail.com

УДК 004.652

**ОБЗОР ФРЕЙМВОРКОВ ДЛЯ ОТОБРАЖЕНИЯ КАРТОГРАФИЧЕСКИХ ДАННЫХ
НА МОБИЛЬНЫХ УСТРОЙСТВАХ ПОД УПРАВЛЕНИЕМ iOS**

Сверчков В.А.

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

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе рассмотрены основные методы отображения картографических данных на устройствах под управлением iOS. Материалы исследования можно использовать для изучения вопросов выбора инструментов разработки под мобильную платформу.

Ключевые слова: мобильная разработка, iOS, картографические сервисы, MapKit, GoogleMaps.

Мобильные приложения, использующие картографические данные получили широкое распространение в различных сегментах рынка. Появилось множество B2C-сервисов, решающих каждодневные проблемы пользователей, от поиска нужного объекта на карте до удобной организации перевозок без необходимости вызова такси по телефону. Помимо этого, в B2B-сегменте открылись широкие возможности для оптимизации и автоматизации бизнес-процессов. Логистические задачи получили наглядную визуализацию, стала возможна реализация систем мониторинга автотранспорта в реальном времени. Государственный сектор реализует множество сервисов, доступных благодаря наличию возможности цифровой картографии: кадастровый учет, мониторинг экологической ситуации, общественного транспорта и многие другие [1, 2].

Вопросу удобного для пользователя и эффективного с технической точки зрения отображения картографических данных уделяется большое внимание. На iOS существует множество фреймворков для показа и взаимодействия с картами. Перед проведением обзора фреймворков необходимо изучить технологии, используемые при отображении карт и функционал, необходимый при работе с картами.

При отображении любой карты, прежде всего, необходима отрисовка слоя с изображением местности. Для реализации данного функционала используется два подхода: с использованием растровой графики и с использованием векторной графики. Рассмотрим их подробнее.

Для показа карты в растровом формате, на сервере сначала формируются изображения для различного масштаба отображения. После полученные изображения разбиваются на мелкие квадратные фрагменты фиксированного размера, так называемые «мозаичные слои» (например, 256×256 пикселей). Когда пользователь выбирает область для отображения на своем мобильном устройстве, то он получает набор изображений с нужным масштабом, достаточный для показа выбранной области на карте. При изменении выбранной области запрашивается новая порция изображений, которая подставляется на фон, создавая у пользователя ощущение непрерывного перемещения по полотну карты. Как правило, фреймворки реализуют кэширование запрошенных изображений в памяти устройства для уменьшения нагрузки на сервер и интернет-соединение.

Сильными сторонами такого подхода являются:

- широкая распространенность использования данного метода в готовых картографических решениях;
- сравнительно низкие требования к техническим характеристикам клиентского устройства;
- возможность отображения спутниковых данных.

При этом существует и ряд недостатков:

- высокая нагрузка на интернет-канал;
- большой объем данных для хранения на мобильном устройстве;
- отсутствие возможности внесения изменений непосредственно на слой с изображением местности;
- дискретность увеличения масштаба;
- при внесении изменений на карту на сервере необходимо пересоздание изображений карт.

Векторный подход предполагает иной способ реализации функционала показа карт на устройстве: сервер присылает набор данных об отображаемых на карте объектах, а их отрисовка происходит непосредственно на клиенте.

Подход имеет следующие преимущества:

- более низкое потребление интернет-трафика;
- возможность тонкой настройки детализации отображения тех или иных данных при различных уровнях увеличения изображения;
- возможность моментальной модификации отображаемых данных;
- лучший пользовательский опыт в сравнении с растровыми картами.

Из недостатков стоит отметить невозможность показа спутниковых изображений и значительно более высокую нагрузку на клиентское устройство.

Помимо показа слоя с изображением местности, как правило, в приложении необходимо отображать дополнительные элементы, обозначающие различные объекты и области на карте. Данные объекты имеют следующую классификацию:

- маркеры – одиночные объекты на карте, например, магазины или транспортные средства. Как правило, подразумевают возможность отклика на пользовательское нажатие;
- линии – чаще всего используются при построении маршрута;
- полигоны – области на карте, имеющие границу и заливку.

В зависимости от предназначения приложения, вышеупомянутые элементы могут занимать большое количество памяти. Для оптимизации реализуется механизм переиспользования экземпляров объектов: как только объект выходит за пределы видимой области карты, то он может быть использован для отображения другого объекта, появившегося в области видимости. Также при необходимости отображения

большого количества маркеров используется кластеризация – объединение множества маркеров в один. При изменении масштаба маркер может либо стать частью ближайшего кластера, либо кластер может распасться на отдельные маркеры.

Для отображения картографических данных на платформе iOS существует фреймворк от компании Apple, который называется MapKit. Он предоставляет собой гибкий программный интерфейс для отображения растровых карт с возможностью изменения основного слоя на нужный, без обязательной привязки к картографическому сервису от компании-производителя. Фреймворк имеет широкие возможности для создания дополнительных элементов на карте, а также возможность построения и отображения подробных маршрутов для различных способов передвижения с использованием сервиса Apple. Стоит отметить, что компания не устанавливает лимитов на доступ данным карт и не запрашивает за них плату. Из недостатков стоит отметить специфичность API и отсутствие реализаций данного фреймворка на других платформах, что может увеличить срок разработки программных продуктов, если один и тот же разработчик создает приложения для разных платформ.

Компания Google также предоставляет свою реализацию карт для iOS – GoogleMaps. Решение имеет схожий API для поддерживаемых платформ, который также будет привычен web-разработчикам. Google в течение продолжительного времени развивает свой картографический сервис и, как правило, имеет более подробные данные об объектах на карте, чем Apple. Существенными недостатками являются отсутствие возможности использования сторонних картографических сервисов, наличие ограничений и платный доступ к данным по достижении определенного лимита.

Одной из реализаций векторных карт является продукт компании Mapbox. Сервис предоставляет карты в различных стилях и предлагает создавать свои, что может быть использовано для лучшей интеграции карт в дизайн мобильного приложения. Благодаря векторному рендерингу карт изменение масштаба происходит плавнее, что также положительно влияет на пользовательский опыт. Продукт имеет ограничение на количество бесплатных запросов в месяц, после которого необходим переход на платный тариф сервиса.

Все перечисленные инструменты обладают функционалом для отображения различного рода объектов на карте и манипуляций с ними.

Таким образом, на сегодняшний день существует множество доступных инструментов для добавления карты в мобильное приложение. В зависимости от своих требований и ограничений можно найти фреймворк, удовлетворяющий требованиям поставленной задачи.

Литература

1. Apple MapKit API Reference [Электронный ресурс]. – Режим доступа: <https://developer.apple.com/reference/mapkit>, своб.
2. Mapbox iOS SDK [Электронный ресурс]. – Режим доступа: <https://www.mapbox.com/ios-sdk>, своб.



Сергейчук Николай Андреевич

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: nicknixer@gmail.com

**Готская Ирина Борисовна**

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, д.педагог.н.,
профессор
e-mail: iringot@mail.ru

УДК 004.67

ОБЗОР ВОЗМОЖНОСТЕЙ СОВРЕМЕННЫХ СИСТЕМ ВЕБ-АНАЛИТИКИ**Сергейчук Н.А., Готская И.Б.****Научный руководитель – д.педагог.н., профессор И.Б. Готская**

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

В работе проведен обзор возможностей современных систем веб-аналитики, приведены основные способы их применения. Выделены различия современных систем, таких как GoogleAnalytics и Яндекс.Метрика. Приведены основные показатели и метрики, которые учитываются этими системами. Сформулированы выводы о пользе данных систем при продвижении сайта.

Ключевые слова: IT, ИТ, Интернет, веб, веб-аналитика, GoogleAnalytics, Гугл Аналитикс, ЯндексMetrika, Яндекс.Метрика.

Использование сети Интернет приобрело массовый характер, он является основным источником новостей, средством для общения, а также позволяет быстро получить доступ к нужной информации. В бизнесе также широко используют сеть Интернет, многие компании имеют свои собственные сайты, которые позволяют решать определенные бизнес-задачи.

Сфера бизнеса, при которой 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.

Популяризировать интернет-ресурс можно при помощи различных средств, наиболее используемыми являются реклама на сайтах и продвижение в поисковых системах с использованием методов поисковой оптимизации. Эффективность используемых средств в продвижении ресурса могут определять системы веб-аналитики. Вопрос использования веб-аналитики при продвижении и на всем жизненном цикле интернет-ресурса является одним из актуальных в сфере современных интернет-технологий.

Однако на современном рынке множество различных систем веб-аналитики, и выбрать систему для решения своих задач становится непросто. Исходя из этого, возникает необходимость в анализе возможностей современных систем веб-аналитики, которые предназначены для мониторинга посещаемости интернет-ресурсов. Эти системы собирают данные о каждом пользователе (посетителе), на основании этих данных определяется аудитория, производится анализ поведения каждого посетителя. Результаты анализа оказывают помощь как в принятии решения по дальнейшему развитию интернет-ресурса и увеличению его посещаемости, так и в оценке эффективности проведенной рекламной компании.

Следует отметить, что возможности современных систем веб-аналитики также позволяют автоматически анализировать мнения и оценки пользователей веб-ресурсов. Администраторам веб-ресурсов это помогает понять, полезна ли та или иная информация, размещенная на их ресурсе. Это достигается путем анализа сообщений пользователей. Например, по смайлам, содержащимся в сообщениях, можно получить оценку отношения пользователя к содержимому ресурса [1].

Существует два основных вида инструментов сбора данных:

1. счетчики. Реализуются при помощи кода, встраиваемого в веб-страницу и следящего за поведением пользователя;
2. лог-анализаторы. Реализуются при помощи программ, работающих с данными из лог-файлов сервера.

Эти два вида инструментов могут использоваться в комплексе. Первый вид инструментов сбора данных чаще используется в современных системах веб-аналитики.

Используя собранные данные, системы веб-аналитики предоставляют показатели и метрики посетителей сайта. Существуют общие ключевые показатели, присущие многим системам веб-аналитики, и показатели, особенные для каждой системы. В таблице представлены характеристики общих ключевых показателей.

Таблица. Характеристики общих ключевых показателей

Показатель	Характеристика
Посещения (трафик, визиты)	Показывает информацию о количестве просмотров страниц и уникальных посетителей за определенный промежуток времени.
Источник трафика	Показывает информацию о посещениях из определенного источника, а также группирует их по категориям: сайты, социальные сети, поисковые системы, рекламные сети, прямые заходы и др.
Глубина просмотра	Показывает информацию о количестве страниц, просмотренных за посещение каждым посетителем, а также средний показатель по всем посетителям.
Новые посетители	Показывает процент новой аудитории сайта. Он рассчитывается как отношение новых посещений к посещениям в общей сумме. Высокий процент данной метрики говорит о том, что сайт не удерживает старую аудиторию.
Просмотры страниц	Показывает информацию о количестве обращений к каждой странице сайта, а также совокупное количество обращений ко всем страницам.
Время на сайте	Показывает среднее время, которое пользователь проводил на сайте. Малое количество времени может указывать на плохое качество предоставляемого контента.
Отказы	Показывает процент посетителей, которые сразу покинули сайт после визита.

Кроме данных метрик, системы веб-аналитики предоставляют информацию о демографической принадлежности посетителей, произведенных действиях посетителем, а также по другим собственным показателям, присущим каждой системе веб-аналитики.

Самыми популярными и современными системами веб-аналитики и составления статистики посещаемости сайтов являются GoogleAnalytics и Яндекс.Метрика (Яндекс.Метрика). Они по отношению друг к другу, как и по отношению к другим системам, имеют свои преимущества и недостатки [2]. Основные различия заключаются в особенностях, присущих каждой из систем.

Отличительными особенностями Яндекс.Метрики являются:

1. интерфейс. Он универсален и интуитивно понятен. На главной странице можно размещать виджеты с любым выбранным показателем и настраивать их вид;
2. обновление данных. Они обновляются в режиме реального времени с интервалом 5 мин, это очень маленький показатель среди таких систем;
3. СМС-отчет. Пользователь имеет возможность подключить СМС-информирование, если с сайтом случаются технические неполадки или перебои;
4. карта кликов. Это главная особенность данной системы, с ее помощью можно отследить самые привлекательные элементы страницы для пользователя.

Отличительными особенностями GoogleAnalytics являются:

1. гибкая настройка фильтров. Система предоставляет большое количество настроек фильтров и режимов работы, однако это усложняет интерфейс;
2. возможность работы с многоканальными последовательностями и социальными сетями;
3. стандартный и улучшенный режимы сегментации трафика.

Анализируя отличительные особенности, становится понятно, что различия несущественные и во многих случаях не будет разницы при выборе между GoogleAnalytics и Яндекс.Метрика, так как они примерно одинаковы по функциональным возможностям и выполняют все функции, которые должны выполнять современные системы веб-аналитики.

Таким образом, GoogleAnalytics и Яндекс.Метрика являются отличными современными инструментами для аналитики сайтов [3–5].

Литература

1. Ихсанов А.Р. Система сбора и анализа статистики пользовательских сообщений на web-сайтах [Электронный ресурс]. – Режим доступа: <http://technology.snauka.ru/2016/05/10043>, своб.
2. Верховцева О.Н. Показатели эффективности маркетинговой деятельности в Интернете [Электронный ресурс]. – Режим доступа: <http://cyberleninka.ru/article/n/pokazateli-effektivnosti-marketingovoy-deyatelnosti-v-internete>, своб.
3. Савельева И.П., Никулин Д.Н. Оценка эффективности интернет-рекламы с помощью систем веб-аналитики // Вестник ЮУрГУ. Серия: Экономика и менеджмент. – 2014. – № 3. – С. 99–105.
4. Егорова И.Н., Кадушкевич О.Н. Методика эффективного использования инструментов googleanalytics // Журнал SCIENTERISE. – 2016. – № 2(18). – С. 40–44.
5. Егорян Л.Б. Новые области применения web-аналитики: социальная сфера и блогосфера // Аудитор. – 2015. – № 6. – С. 43–48.



Серебренникова Ирина Олеговна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: irina_serebrennikova@mail.ru

УДК 373.1

**ЭЛЕКТРОННЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕСУРСЫ ДЛЯ ОБЩЕГО ОБРАЗОВАНИЯ:
СОВРЕМЕННОЕ СОСТОЯНИЕ, ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ**

Серебренникова И.О.

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 осуществлен аналитический обзор состояния рынка электронных образовательных ресурсов для общего образования: проведен анализ современного состояния, анализ проблем в сфере электронных образовательных ресурсов и анализ перспектив развития электронных образовательных ресурсов для общего образования.

Ключевые слова: электронный образовательный ресурс, ЭОР, образование, информационные продукты, образовательный процесс, эффективность.

В наше время учебный процесс требует расширения арсенала средств обучения для реализации максимальных возможностей современной информационной среды. В частности, средств электронного обучения, связанных с электронными образовательными ресурсами (ЭОР), под которыми сегодня понимается весь спектр информационных объектов, а также их комбинации, информационные среды, поддерживающие учебную деятельность, электронные учебники, курсы и другие средства, воспроизводимые на электронных устройствах [1].

Электронные образовательные ресурсы значительно упрощают проведение занятий, мониторинг результатов обучения и собственно обработку этих результатов. Кроме того, целью ЭОР является не только предоставление учебного материала, но и возможности его практического применения, и проведение аттестации в ходе процесса обучения.

По результатам мониторинга использования ЭОР среди преподавателей России были сделаны выводы о состоянии ЭОР в наше время, их практической пользе и заинтересованности в их использовании как преподавателей, так и обучающихся с использованием данных технологий.

Согласно данным федерального проекта «Развитие электронных образовательных Интернет-ресурсов нового поколения, включая культурно-познавательные сервисы, системы дистанционного общего и профессионального обучения (e-learning), в том числе для использования людьми с ограниченными возможностями» большинство преподавателей используют на своих занятиях средства электронного обучения постоянно (71,2%).

Так, в 2012 году доля учителей, использующих ЭОР в учебной деятельности, была 25%, в последующих годах она увеличилась практически вдвое (рис. 1).

Анализируя данные опроса преподавателей о качестве и эффективности ЭОР можно заметить большую заинтересованность в использовании ЭОР на занятиях. Но, ввиду того, что решение о методе проведения занятий зависит от преподавателя, то важным является его информационная грамотность в сфере информационных технологий [2].

Так, например, видна зависимость качества занятий с применением ЭОР от степени владения преподавателем технологиями. Она наблюдается, если спросить преподавателей о возможных причинах снижения эффективности использования ЭОР.



Рис. 1. Мнения педагогов об эффективности ЭОР по сравнению с традиционными средствами обучения в зависимости от опыта их использования (%)

Также факторами снижения эффективности является оснащенность учебных аудиторий необходимым оборудованием, недостаточное количество времени на разработку эффективного ЭОР, а также психологические зажимы в использовании современных технологий. Оценка эффективности ЭОР значительно изменяется с увеличением частоты использования данных в преподавательской практике. Это объясняется формированием необходимых умений преподавателя в разработке ЭОР, его корректировке и оптимальном применении в образовательной деятельности.

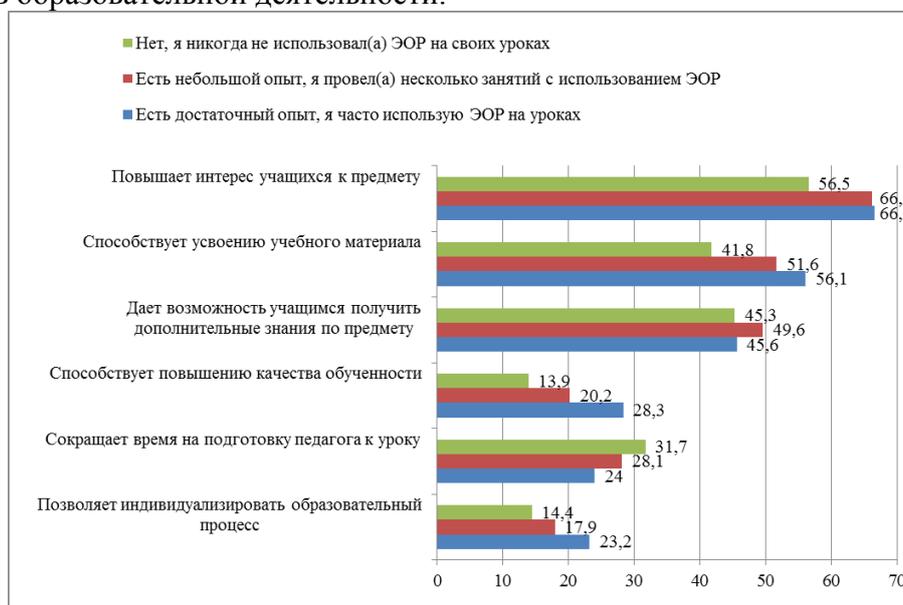


Рис. 2. Мнения педагогов об эффективности ЭОР в зависимости от опыта их применения (%)

Из полученных данных (рис. 2) возможно сделать вывод о возможностях ЭОР и возникающих проблемах при их использовании.

Нет сомнений в том, что электронные образовательные ресурсы очень эффективны при грамотном использовании. Многочисленные возможности взаимодействия между

преподавателем и учеником позволяют своевременно проводить мониторинг освоенного материала и корректировать дальнейшее обучение [3].

С использованием ЭОР значительно повышается интерес учеников к учебному материалу. Этому способствует использование различных интерактивных материалов и возможность прохождения курса в дистанционном формате.

Из отрицательных сторон ЭОР можно выделить сложность подготовки учебного материала. Немаловажно здесь квалификация преподавателя. Только с опытом использования ЭОР приходит понимание, как сделать курс интересным и эффективным. Важно проводить ознакомительные курсы с преподавателями, где они могут ознакомиться с возможностями и самой технологией.

И, как ранее говорилось, одной из главных проблем в использовании ЭОР является техническое обеспечение образовательных учреждений. Разработка стандартов для проведения дистанционного обучения значительно упростила подготовку оборудования и учебных аудиторий, но в основном главное – это достаточное финансирование образовательных учреждений и целесообразность использования данной технологии [4].

На данный момент происходит переход к дистанционным программам обучения. Сейчас можно заметить множество курсов с дистанционной поддержкой, где ученик, пропустив занятие, может так же, как и остальные ученики изучить лекционный материал и пройти тестирование, за результатами которого будет наблюдать преподаватель, который оценит прохождение материала также дистанционным способом.

Наиболее распространенными системами электронного обучения на данный момент являются обучающая система «Moodle» и сервис «Электронный дневник и журнал». Их возможности позволяют преподавателям донести материал до учеников, а ученикам, в свою очередь, пройти его, усвоить и пройти аттестацию. В общеобразовательных заведениях это еще и хорошая возможность родителям наблюдать за учебными успехами своих детей.

В результате можно сказать, что на современном этапе развития очень многое зависит от самих преподавателей, их квалификации и опыта работы с ЭОР. В дальнейшем, конечно, будет наблюдаться рост использования дистанционных систем обучения в связи с ростом опыта использования технологий в данной области. Дальнейшее знакомство с технологией и информатизация преподавателей приведет к значительному росту эффективности использования электронного образования, а постепенное технологическое обеспечение образовательных учреждений – и к возможностям проведения дистанционного обучения.

В последующей работе над данной темой следует рассмотреть различные технологии для создания ЭОР, возможности для составления грамотных и эффективных программ обучения, а также рассмотрение существующих стандартов для дальнейшего создания курсов дистанционного образования.

Литература

1. Опыт использования ЭОР [Электронный ресурс]. – Режим доступа: <http://eor-np.ru/node/90>, своб.
2. Бордовский Г.А., Готская И.Б., Ильина С.П., Снегурова В.И. Использование электронных образовательных ресурсов нового поколения в учебном процессе: научно-методические материалы. – СПб.: Изд-во РГПУ им. А.И. Герцена, 2007. – 31 с.
3. Бородин Т.Ф. Применение электронных образовательных ресурсов в образовательном процессе вуза и определение их эффективности // Молодой ученый. – 2014. – № 13. – С. 241–243.
4. Мониторинг ЭОР [Электронный ресурс]. – Режим доступа: <http://eor-np.ru/node/55>, своб.

**Погорелов Виктор Иванович**

Год рождения: 1940

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, д.т.н., профессор

e-mail: pogvic@mail.ru

**Смирнов Денис Юрьевич**

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4108Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: feuway@gmail.com

УДК 004.051

**МЕТОДЫ ОПТИМИЗАЦИИ РАЗРАБОТКИ САЙТОВ С ИСПОЛЬЗОВАНИЕМ
ВЕБ-ПРОГРАММИРУЕМЫХ ФРЕЙМВОРКОВ****Смирнов Д.Ю.****Научный руководитель – д.т.н., профессор Погорелов В.И.**

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе рассмотрены возможности фреймворков для решения различных задач с целью облегчения разработки, выявлены причины их частого использования, рассмотрены их плюсы и минусы.

Ключевые слова: веб-фреймворк, YII, AngularJS.

Постоянный рост требований к созданию новых веб-приложений приводит к увеличению затраченного времени и нагрузки на процесс разработки. Все чаще эти факторы провоцирует разработчиков применять все более современные средства разработки, например, различные веб-фреймворки [1–4].

Фреймворк – это программное средство для облегчения процесса разработки, упрощающее создание и поддержку сложных и высоконагруженных проектов. Веб-фреймворки можно считать платформой при создании сайтов, бизнес-приложений и веб-сервисов. Обычно сам фреймворк включает в себя только базовые программные модули, все же остальные необходимые для конкретного проекта компоненты создаются разработчиками на их основе.

В последнее время базовой платформой для создания веб-приложений стали РНР-фреймворки. Сейчас сложно представить разработку веб-приложений, их сопровождение и модернизацию без использования данного программного средства. Они обеспечивают основную структуру программы, экономят большое количество времени, уменьшают нагрузку на процесс разработки, решают проблему повторяющегося кода.

Большинство применяемых РНР-фреймворков используют архитектуру MVC. Model View Controller (MVC) – это архитектурный шаблон проектирования, позволяющий разделить бизнес-логику приложения от его пользовательского интерфейса. При этом веб-приложение разделяется на три основных компонента: Model – часть взаимодействующая с базой данных, View – часть представляющая то, что непосредственно видит пользователь

(графический интерфейс приложения), Controller – это часть занимается контролем и управлением всеми составляющими и данными приложения (рисунок).

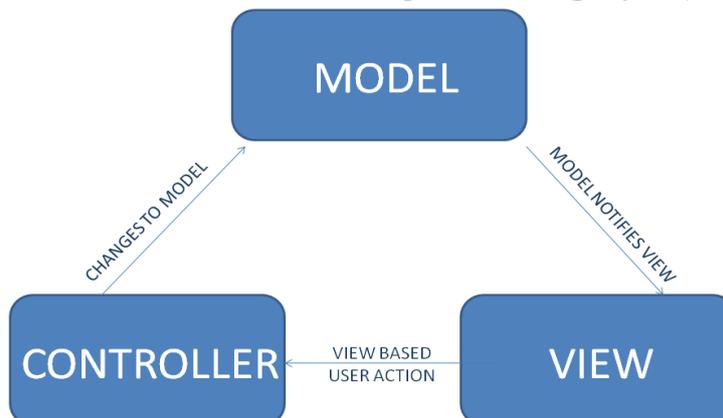


Рисунок. Концепция MVC

Для того чтобы более детально ознакомиться со всеми преимуществами и недостатками использования веб-фреймворков, рассмотрим одни из самых популярных.

YII – это построенный на компонентной структуре PHP-фреймворк, использующийся для разработки практически любых веб-приложений. Также использует архитектурный шаблон проектирования MVC.

Его особенностями можно назвать:

- высокую производительность;
- легковесность;
- возможность перехвата и обработки ошибок;
- использование AJAX и интеграцию с jQuery;
- поддержку REST и автоматического тестирования;
- встроенную поддержку интернационализации;
- большое русскоязычное сообщество;
- кеширование страниц и отдельных фрагментов;
- поддержку тем оформления.

Существенным недостатком можно назвать то, что данный фреймворк достаточно молодой, и в любое время в нем могут найти критические ошибки.

AngularJS – это клиентский MVC-фреймворк, написанный на JavaScript, выполняемый в веб-браузере, предназначенный для облегчения создания современных клиентских веб-приложений, использующих технологии AJAX. Данный фреймворк работает напрямую с HTML, который содержит различные дополняющие пользовательские атрибуты, описываемые директивами, связывает ввод или вывод области страницы с моделью в виде обычных переменных языка JavaScript, значения которых либо задаются самостоятельно, либо получают из JSON-данных.

Приложения, написанные с использованием AngularJS:

- легко расширяемы;
- легко сопровождаемы;
- поддаются тестированию;
- поддерживают стандарты.

Несмотря на многочисленные особенности, AngularJS обладает и несколькими недостатками:

- сложен в освоении (особенно трудно поддается данный фреймворк тем, кто привык работать с jQuery);
- сильно замедляется работа приложения, когда используются более 2000 вотчеров (слушателей событий);
- нет обратной совместимости со второй версией данного фреймворка.

Можно сказать, что создание веб-приложений на фреймворках позволяет добиться простоты сопровождаемости проекта. Относительно просто осуществляется реализация различных бизнес-процессов, проекты становятся легко масштабируемыми и модернизируемыми. Приложения, написанные на фреймворках, зачастую гораздо произвольнее, быстрее и обладают высокой сопротивляемостью к большим нагрузкам. Все эти перечисленные факты соответствуют требованиям к созданию сложных современных веб-приложений.

Литература

1. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – Изд-во: Питер, 2016. – 768 с.
2. Sklar D. Learning PHP: A Gentle Introduction to the Web's Most Popular Language. – O'Reilly Media, 2016. – 416 p.
3. Козловский П., Дарвин П.Б. Разработка веб-приложений с использованием AngularJS. – Изд-во: ДМК Пресс, 2014. – 394 с.
4. Уайнсет Дж., Сафронов М. Разработка веб-приложений в YII 2. – Изд-во: ДМК Пресс, 2015. – 392 с.



Соболева Александра Антоновна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4206
Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: sasha-soboleva@mail.ru



Перепелица Филипп Александрович

Год рождения: 1977

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, ст. преподаватель
e-mail: perepelitcafa@niuitmo.ru



Шалобаев Евгений Васильевич

Год рождения: 1947

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, к.т.н., доцент
e-mail: kpd@limtu.ru

УДК 004.94

БЫСТРОЕ ПРОТОТИПИРОВАНИЕ СРЕДСТВАМИ АДДИТИВНЫХ ТЕХНОЛОГИЙ

Соболева А.А., Перепелица Ф.А., 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 настоящее время быстрое прототипирование пользуется большой популярностью. На сегодняшний день известно более 10 технологий послойного создания объектов, и с годами их становится все больше. В работе рассмотрены преимущества и недостатки быстрого прототипирования средствами аддитивных технологий.

Ключевые слова: аддитивные технологии, прототипирование, АМ-технологии, полиамид, 3D-принтер.

Аддитивные технологии – это послойное изготовление различных изделий, основанное на компьютерной 3D-модели. Процесс изготовления детали постепенный, поэтому его зачастую называют «выращиванием». Главное отличие данного процесса от традиционного производства состоит в том, что при стандартном производстве для начала изготавливается модель, от которой отсекается лишнее, либо происходит деформация данной модели, а с помощью аддитивных технологий из амфорного расходного материала сразу получается необходимое изделие [1].

Быстрое прототипирование – изготовление прототипа модели в максимально короткие сроки. Данная задача практического применения АМ-технологий является основной. Говоря «прототип» в данном случае, подразумевается большой спектр значений. На этапе разработки изделия необходимо быстро получить прообраз конечного результата модели.

Разработка прототипа важна для рассмотрения некоторых факторов:

- изучение геометрических форм детали;
- оценка эргономических качеств;
- проверка собираемости;
- правильность компоновки.

Главным преимуществом «быстрого» изготовления детали является то, что существенно сокращаются сроки разработки объекта. В данном случае прототип – это модель, предназначенная для каких-либо испытаний или для предварительной проверки функциональности изделия. Прототипы бывают масштабными. Иногда прототипы используются для поиска интересных дизайнерских решений, например, в цветовой гамме, или в различных других нюансах.

Одним из главных преимуществ использования аддитивных технологий является возможность изготавливать функциональные прототипы, не используя дорогостоящие технологические оснастки. Можно рассмотреть в качестве примера блок цилиндров двигателя внутреннего сгорания, которые могут быть использованы в проведении важных моторных испытаниях. Данный пример является одним из методов быстрого прототипирования, в котором не используются массивные изделия из дерева или металла [2].

Быстрое прототипирование средствами аддитивных технологий используют в основном на первоначальных стадиях проектов для того, чтобы рассмотреть геометрический образ объекта. Данная стадия проходит без определения прочности, гибкости, фактуры и других свойств материала. Материал выбирается из предложенных модельных материалов, который наибольшим образом подходит для поставленной цели визуализации. Бывает, что свойства позволяют провести испытания прототипа на функциональную устойчивость.

Для создания прототипов чаще всего используют недорогие 3D-принтеры. На сегодняшний день существует большое количество различных технологий, в результате сравнительного анализа можно выбрать наиболее подходящую технологию по цене принтера и материалов, качеству и времени разработки [3].

Шероховатость поверхности прототипа зависит от ориентации модели на рабочей платформе при построении. Если надо построить плоское изделие в хорошем качестве на боковых поверхностях объекта, то необходимо сориентировать под углом к плоскости платформы. В случае расположения детали горизонтально, возможно появление низкой шероховатости на горизонтальной поверхности.

Для того чтобы создать изделие с хорошим качеством поверхности, обязательно надо соблюсти один из важнейших параметров – создание качественной исходной компьютерной 3D-модели. Данная модель – это 3D-поверхность, представленная в виде замкнутой сетки. Создание

именно этой сетки отвечает за шероховатость поверхности. Данная сетка должна быть построена качественно, если использовать трехмерные модели низкого качества, при построении физической модели возможно появление шероховатости. На рисунке можно наблюдать, что создание сетки с малым количеством сегментов ведет к изменению даже геометрической формы. Если сравнить рисунок, а, и рисунок, б, на котором представлена та же модель, только с наибольшим количеством сегментов сетки, можно сделать вывод о влиянии данного фактора на физическую модель. Данное явление может сильно повлиять на мнение о качестве принтера и обо всей технологии в целом, что не является правдоподобным мнением.

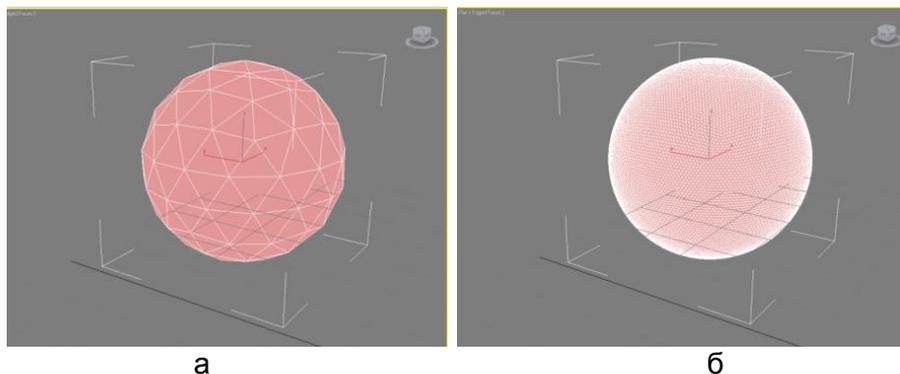


Рисунок. Примеры плохого создания сетки (а) и правильного создания сетки (б)

Одним из направлений быстрого прототипирования является производство «быстрой оснастки» [4]. Данный вид производства применяют перед изготовлением серии деталей не из модельных материалов, а из промышленных пластиков. Также производство «быстрой оснастки» может пригодиться для изготовления штучной или серии продукции с малым количеством экземпляра (до ста штук).

С использованием АМ-технологии значительно сократилось время производства, а также возможны моторные испытания данного прототипа, созданного из полиамида. В наше время появляется множество новых высокопрочных и термостойких модельных материалов, что дает большие возможности для создания функциональных прототипов.

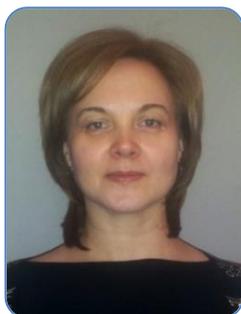
Наиболее значимым недостатком технологии быстрого прототипирования на сегодняшний день является то, что 3D-принтеры строят изделия только из модельных материалов. Данные материалы не всегда будут соответствовать функциональным требованиям той или иной модели.

Подводя итог, можно сказать, что главными преимуществами быстрого прототипирования средствами аддитивных технологий является:

- меньшая трата финансовых средств;
- экономия большого количества материалов;
- увеличение скорости создания прототипов.

Литература

1. Перспективы развития аддитивных технологий [Электронный ресурс]. – Режим доступа: http://rtb-consult.ru/reviews-catalog/additive_technologies/, своб.
2. Зленко М.А., Нагайцев М.В., Довбыш В.М. Аддитивные технологии в машиностроении. Пособие для инженеров. – М.: ГНЦ РФ ФГУП «НАМИ». 2015. – 220 с.
3. Гибсон Я., Розн Д., Стакер Б. Технологии аддитивного производства. Трехмерная печать, быстрое прототипирование и прямое цифровое производство. – М.: ТЕХНОСФЕРА, 2016. – 656 с.
4. Шишковский И.В. Основы аддитивных технологий высокого разрешения. – СПб.: Питер, 2015. – 348 с.



Соловьева Наталья Геннадьевна

Год рождения: 1970

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра управления и права, группа № S4215

Направление подготовки: 38.04.02 – Менеджмент

e-mail: solovjova.n@mail.ru

УДК 34.08

ИССЛЕДОВАНИЕ И РАЗРАБОТКА МЕТОДОВ МОТИВАЦИИ ПЕРСОНАЛА ДЛЯ УЛУЧШЕНИЯ КАЧЕСТВА РАБОТЫ ОРГАНИЗАЦИИ

Соловьева Н.Г.

Научный руководитель – к.ю.н., доцент Воронина М.В.

Работа является второй частью исследования и разработки системы мотивирующих методов для улучшения качества работы организации, занимающейся поставкой и обслуживанием программного обеспечения. Проведена практическая работа по выявлению точек разрыва (несоответствий) в системе качества, проанализировано состояние коллектива в плане удовлетворенности работой и степени вовлеченности в развитие организации, выявлена их взаимосвязь. В заключение сделаны рекомендации по соответствующим методам мотивации.

Ключевые слова: методы мотивации, оценка качества, жизненный цикл комплекса «продукт-услуга», вовлеченность персонала, индекс вовлеченности.

Под понятием «качества» в современном менеджменте подразумевается степень соответствия совокупности присущих характеристик комплекса «продукт-услуга» требованиям к ней [1]. Современное понимание концепции управления комплексным качеством TQM предполагает идею непрерывного улучшения всего, что организация делает (что основано на принципах Деминга) и ориентацию на потребителя [2]. Эти принципы должны внедряться на всех этапах жизненного цикла производства, поставки и обслуживания программного обеспечения (ПО) (рис. 1).



Рис. 1. Петля качества комплекса «продукт-услуга»

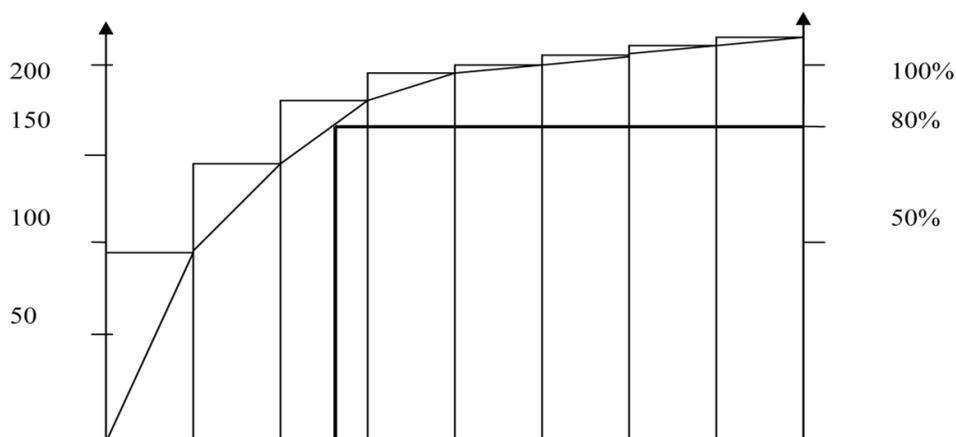


Рис. 2. Диаграмма Парето, построенная на основе собранных данных

Менеджментом компании было принято решение производить управление качеством на основе «модели разрывов» – Gap Model (впервые введено Zeitman, Berry, Parasuraman, 1988), которая позволяет определить основные классы ошибок в бизнес-процессе любой компании. Обратную связь по качеству компания получает в виде отрицательных отзывов клиентов – по телефону в отдел качества, либо при непосредственном общении специалистов отдела обслуживания. Период наблюдений составляет 1 квартал. За этот период было зарегистрировано 1021 обращение клиентов. Из них жалобы составляют 208. По полученным данным построена диаграмма Парето [2] (рис. 2). Из построенного графика можно выявить те факторы, на долю которых приходится в сумме 80% всех жалоб.

Делаем вывод, что самыми значимыми причинами жалоб являются следующие:

1. ПО дает сбой при работе (не доведение до безотказного продукта);
2. неадекватное общение консультантов отдела обслуживания;
3. несоблюдение сроков выполнения заказа.

В ходе работы изучены существующие теории мотивации [3]. Был проведен анализ существующей в организации на данный момент системы мотивации (материальная и нематериальная части), выделена как проблема сложная система расчета заработной платы, построенная на коэффициенте трудового участия, а также изучены путем проведения сплошного анкетирования настоящее положение дел с удовлетворенностью работой и вовлеченностью персонала. В соответствии с ГОСТ [4], основными принципами менеджмента являются: лидерство руководителя, вовлечение работников, постоянное улучшение, принятие решений, основанное на фактах. В связи с этим состояние вовлеченности персонала напрямую влияет на качество работы организации. Расчет индекса вовлеченности был произведен по методике, предложенной Е.А. Скриптуновой [5]. В настоящей работе не представлялось возможным подробно рассмотреть данную методику, приведены лишь шкалы, по которым оценивается данный параметр (табл. 1 и 2).

Таблица 1. Шкалы составляющих вовлеченности и веса каждой из них

Показатель вовлеченности	Шкала вопросов (баллы)	Количество вопросов	Индекс по показателю (баллы)
Вовлеченность в решение корпоративных задач	0–6	8	0–48
Вовлеченность в рабочий процесс	0–4	11	0–44
Инициативность	0–1	8	0–8
Вовлеченность работника в целом			0–100

Таблица 2. Оценка уровня вовлеченности по полученному индексу

Уровень вовлеченности	Индекс вовлеченности
Низкий	0–20
Ниже среднего	20–40
Средний	40–60
Выше среднего	60–80
Высокий	80–100

В итоге работы для руководства были представлены результаты оценки вовлеченности по отделам и в целом по предприятию. Они приведены в табл. 3.

Таблица 3. Результаты исследований вовлеченности

Исследуемый отдел	Индекс вовлеченности
Отдел продаж	57
Технический отдел	45
Отдел обслуживания	72
Всего по компании	58

Таким образом, коллектив в целом вовлечен в дела предприятия на среднем уровне, что свидетельствует о незадействованных резервах по мотивации. По итогам работы авторами предложены меры по стимулированию в части качества. Некоторые из них:

1. упростить формулы расчета заработной платы таким образом, чтобы каждый выигрывал от общего результата в области качества, а не только своего личного (премиальный фонд, выплачиваемый сотрудникам на уровне отделов, поставить в обратную зависимость от количества несоответствий);
2. организовать на предприятии так называемые «кружки качества», на которых регулярно (например, раз в 2 недели) сотрудники могли бы обмениваться опытом разрешения сложных рабочих ситуаций;
3. создать постоянно действующий внутренний Учебный центр силами своих специалистов, где сотрудники могут получать своевременную консультацию по темам, которые при анкетировании выявлены как сложные. При этом преподающих специалистов поощрять дополнительными оплачиваемыми отпусками;
4. проводить один раз в год оценку уровня вовлеченности персонала по этой же методике, сопоставлять и определять эффективность внедренных мер.

Литература

1. ГОСТ Р ИСО 9001-2015. Системы менеджмента качества. Требования. – Введен 01.11.2015. – М.: Стандартинформ, 2015. – 24 с.
2. Овсянко Д.В. Управление качеством. – СПб.: Высшая школа менеджмента, 2011. – С. 101.
3. Кибанов А.Я. Основы управления персоналом: учебник. – 2-е изд., перераб. и доп. – М.: НИЦ ИНФРА-М, 2014. – 447 с.
4. ГОСТ Р ИСО 9000-2008. Системы менеджмента качества. Основные положения и словарь. – Введен 10.09.2009. – М.: Стандартинформ, 2009. – 26 с.
5. Скриптунова Е.А. Методика расчета индекса вовлеченности персонала [Электронный ресурс]. – Режим доступа: <http://www.axima-consult.ru/stati-04-10-metod-raschet-indeksa-vovl-personala.html>, своб.



Солодовников Алексей Юрьевич

Год рождения: 1986

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4208

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: solo@solo.md

УДК 004.9

ОБЗОР ТЕХНОЛОГИИ SERVICE WORKERS (ПРОДУКТА ЭВОЛЮЦИИ WEB WORKERS) ДЛЯ ОПТИМИЗАЦИИ РАБОТЫ OFFLINE-FIRST ВЕБ-ПРИЛОЖЕНИЙ

Солодовников А.Ю.

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

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе рассмотрены особенности, жизненный цикл и архитектура технологии Service Workers, предназначенная для оффлайновой работы (или работы в условиях плохого соединения) современных (так называемых прогрессивных) веб-приложений.

Ключевые слова: технология Service Worker, прогрессивные веб-приложения, оффлайн.

Проблематика и краткая характеристика предлагаемого решения. Современные веб-приложения, как правило, адаптивны. Это связано с большим ростом популярности мобильных устройств, которые используются для работы с веб-приложениями. Следовательно, перед современными веб-приложениями возникает задача корректной работы в условиях использования на мобильном устройстве при плохом соединении с сетью или даже при отсутствии подключения к сети [1–3].

Этот вопрос решает технология Service Workers, спецификация которой на данный момент еще не окончательная, но реализация уже доступна для браузеров Chrome, Firefox и Opera, разработка ведется для Edge, а компания Apple рассматривает вариант разработки для браузера Safari. Так как технология связана с оптимизацией работы приложения, и не является ключевой технологией для работы приложения, использовать ее можно уже сегодня. Тем более технология считается существенной частью для разработки так называемых прогрессивных веб-приложений (PWA).

Важно знать, что Service Worker представляют довольно мощные средства по управлению данными веб-приложения, поэтому в целях безопасности было решено, что использование Service Worker-ов разрешается только для приложений, использующих протокол HTTPS (HTTP работает только для localhost в целях облегчения процесса разработки).

Ключевой момент – существующая технология Web Worker, которая позволяет исполнять JavaScript в отдельном потоке, без оказания влияния на производительность страницы веб-приложения, при этом Web Worker в целях безопасности не имеет возможности доступа к веб-приложению, и общается с ним посредством сообщений.

Технология Web Worker лежит в основе Service Worker-ов, которые фактически являются промежуточным слоем между страницей веб-приложения и сервером. Этот слой перехватывает запросы клиента на сервер и выполняет определенную логику, в соответствии с которой, как правило, данные предоставляются клиенту моментально (из локальной базы данных или кэша), а последние данные с сервера обновляют приложение только после того как клиент действительно получил их. Разумеется, этот механизм полностью находится в руках разработчика и позволяет указать, какие именно данные должны храниться на клиенте, какие не должны, и какая

информация должна быть предоставлена пользователю по запросу для улучшения пользовательского опыта. Надо понимать, что технология Service Workers динамическая и значительно отличается от простого кэша, который является только лишь статическим хранилищем, используемым Service Worker-ом как одним из средств получения данных.

Архитектура типичного PWA-приложения, использующего Service Workers. Обычно схема взаимодействия PWA с сервером (рис. 1) происходит следующим образом:

- браузер (1) отправляет серверу (4) fetch-запрос;
- Service Worker (2) перехватывает запрос, далее все зависит от внутренней логики Service Worker. Обычно, Service Worker анализирует запрос и пытается получить данные из локального хранилища (5, 6), а в случае отсутствия данных, передает запрос дальше серверу (3, 4), и обновляет локальное хранилище данными с сервера;
- второй и следующие попытки запроса ресурсов будут работать даже если соединения нет, потому что Service Worker достанет их из локального хранилища.

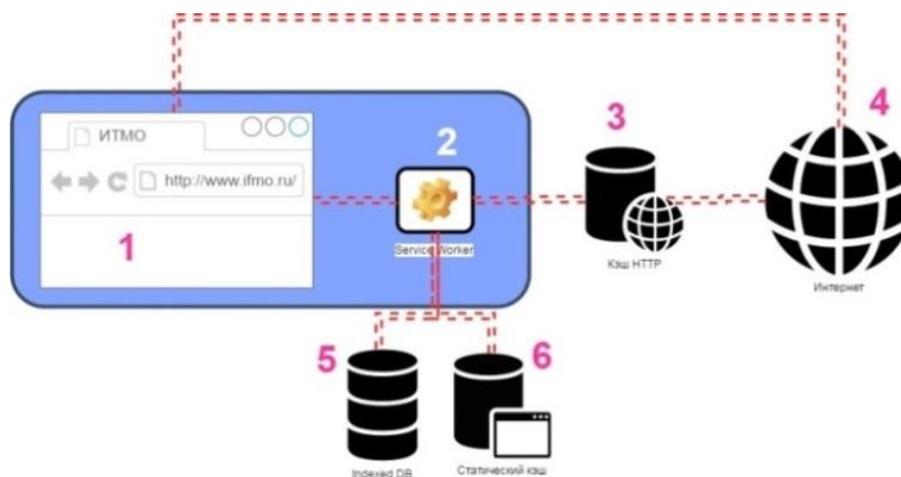


Рис. 1. Схема взаимодействия PWA с сервером: 1 – браузер; 2 – Service Worker; 3 – кэш HTTP; 4 – Internet; 5 – БД IndexedDB; 6 – статический кэш

Статический кэш используется в основном для статических ресурсов. Например, для изображений. IndexedDB (а также любое другое локальное браузерное хранилище вроде LocalStorage или других) используется для хранения данных.

Прямая связь между браузером и сервером (1 \leftrightarrow 4) может быть осуществлена в обход Service Worker-а, например, с помощью технологии WebSocket. Это полезно, если часть данных хранится на сервере (4), а часть – на клиенте (5, 6).

Пример такого сценария – новостная лента. Первый запрос клиента кэширует последние новости в локальном хранилище. Далее пользователь открывает веб-приложение через несколько дней. В этом случае, исходя из логики offline-first приложения, мы показываем клиенту новости из локального хранилища.

В случае если Интернет-соединения нет (или оно плохое и у нас нет ответа с сервера), то на этом сценарий заканчивается. Пользователь видит новостную ленту и может пользоваться приложением. Надо отметить, что локальные данные и статический контент позволяют показать пользователю информацию полностью (текст+графика). Веб-приложение при этом полностью функционально.

Если Интернет-соединение есть, мы получаем от сервера по веб-сокету информацию о последних новостях, обновляем эти данные в локальном хранилище, и показываем пользователю полный список новостей.

Жизненный цикл Service Worker-а. Так как Service Worker – это самостоятельный компонент архитектуры веб-приложения, то и его жизненный цикл также является самостоятельным.



Рис. 2. Жизненный цикл Service Worker-a

Для регистрации Service Worker-a приложение (рис. 2) должно инициализировать его установку (1), которая проходит в фоновом режиме. В процессе установки (1) Service Worker пытается получить и закэшировать ресурсы. Если кэширование проходит успешно, Service Worker считается установленным успешно и активируется (3). В случае какой-либо ошибки (2) Service Worker не будет установлен, соответственно, не будет активирован.

Этап активации (3) Service Worker-a – это самый лучший момент для того, чтобы почистить/обновить старые кэши. После активации Service Worker входит в режим ожидания (4) и контролирует все страницы веб-приложения, которые попадают под его область видимости.

Из режима ожидания Service Worker может попасть либо в рабочий режим перехвата сообщения и выполнения своей внутренней логики, либо завершить свою работу.

Следует отметить, что активированный Service Worker не завершает свою работу до тех пор, пока его использует хотя бы одно приложение. Если новая версия приложения устанавливает новый Service Worker, то и новый, и старый «живут» одновременно до тех пор, пока все страницы, использующие старый Service Worker, не будут закрыты.

Вывод по обзору технологии Service Workers. Разумеется, в случае оффлайна, пользователь не видит последних данных. Тем не менее, сам факт того, что пользователь может пользоваться функционалом приложения, а также имеет доступ к каким-то (в худшем случае, устаревшим) данным – это огромный шаг вперед в области веб-приложений. Возможно, это та самая точка, которая завершит вытеснение десктопных приложений в пользу веб-приложений на рынке программного обеспечения.

С учетом существования на рынке продуктов, которые предоставляют облачные хранилища с возможностью временного хранения данных на клиенте, с последующей синхронизацией и автоматическим разрешением конфликтов, использование технологии Service Workers представляется довольно простым, и в то же время мощным и эффективным способом улучшить пользовательский опыт в области использования новых и уже существующих веб приложений.

Литература

1. Спецификация технологии Service Workers консорциума W3C [Электронный ресурс]. – Режим доступа: <https://www.w3.org/TR/service-workers/>, своб.
2. Прогрессивные веб-приложения [Электронный ресурс]. – Режим доступа: <https://developers.google.com/web/progressive-web-apps/>, своб.
3. Арчибальд Дж., Уэйлс М. Оффлайновые веб-приложения [Электронный ресурс]. – Режим доступа: <https://classroom.udacity.com/courses/ud899>, своб.



Сохина Анна Альбертовна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: sokhina-ann@yandex.ru



Готская Ирина Борисовна

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна,

д.педагог.н., профессор

e-mail: iringot@mail.ru

УДК 004

ОБЗОР ПОПУЛЯРНЫХ СИСТЕМ СБОРКИ ДЛЯ ФРОНТЕНД-ПРОЕКТОВ

Сохина А.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».

В работе приведен обзор наиболее популярных на данный момент систем сборки фронтенда, функционирующих на платформе Node.js, отмечены их основные особенности.

Ключевые слова: сборка проекта, автоматизация сборки, система сборки, фронтенд-разработка, grunt, gulp, webpack, broccoli, brunch.

Еще несколько лет назад в процессе веб-разработки не было необходимости отдельно собирать фронтенд, так как веб-сайты были сравнительно небольшими. Однако в дальнейшем размер кода и его сложность возрастали, и в связи с этим форма, в которой проект разрабатывался, стала значительно отличаться от формы, в которой проект должен демонстрироваться пользователям. Возникли задачи минификации кода, компиляции файлов препроцессоров и их конкатенации.

Поначалу для сборки фронтенда применялись инструменты, использовавшиеся для бэкенда (например, Ant). Однако из-за того, что данные системы изначально не были ориентированы на специфику фронтенда, со временем появились специализированные build-системы, нацеленные конкретно на решение задач во фронтенд-разработке.

Типичная система сборки выполняет следующие функции: конкатенация js- и css-файлов; валидация кода; минификация кода, а также его обфускация (если необходимо); компиляция кода препроцессорных языков; оптимизация изображений; тестирование и др.

Рассмотрим несколько популярных на данный момент build-систем для фронтенда, функционирующих на платформе Node.js:

– Grunt: один из первых инструментов для сборки. На данный момент Grunt имеет большое количество плагинов – 6,070 (по состоянию на январь 2017), тысячи проектов, многомиллионное комьюнити разработчиков, понятная документация, статьи и доклады (в том числе и на русском языке). Рассмотрим пример конфиг-файла Grunt, который осуществляет сборку и обработку CSS:

```
const grunt = require ('grunt');
```

```

grunt.loadMpmTasks ('grunt-sass');
grunt.loadMpmTasks ('grunt-autoprefixer');
grunt.initConfig ({
  sass: {
    dist: {
      files: [{
src: 'dev/*.scss',
expand: true,
dest: '.tmp/styles',
ext: '.css'}]}},
  autoprefixer {
    dist: {
      files: [{
src: '{,*/}*.*css',
expand: true,
cwd: '.tmp/styles',
dest: 'css/styles'}]}},
  watch: {
    styles: {
      files: ['dev/*.scss'],
      tasks: ['sass', 'autoprefixer']}}});
grunt.registerTask('default', ['sass', 'autoprefixer', 'watch']);

```

Из данного примера видно, что примерная последовательность работы Grunt следующая: берется файл из некоторого входного каталога, затем первый плагин считывает и обрабатывает данный файл, и далее сохраняет его во временном хранилище; далее сохраненный файл считывает и обрабатывает уже следующий плагин и также сохраняет во временном хранилище, и т.д. в зависимости от количества плагинов, пока данный файл не будет записан в итоговый каталог с результатами обработки. Промежуточные сохранения файла после обработки каждым плагином требуют некоторых временных затрат, в результате чего Grunt может заметно проигрывать в скорости другим системам сборки на Node.js. Также некоторые разработчики отмечают объемный конфиг-файл Grunt: чтобы настроить простую сборку потребуется файл на сотню строк кода. Однако в силу популярности Grunt и его огромной экосистемы не составит труда найти уже готовый конфиг-файл конкретно под свои задачи [1].

– Gulp. В основе архитектуры системы Gulp лежит понятие потоков в Node.js, что позволяет не сохранять на диск временные файлы, а также запускать задачи параллельно, в отличие от Grunt. Из этого вытекает основное достоинство Gulp – это скорость [2]. Также Gulp имеет более краткий в сравнении с Grunt конфиг-файл, что достигается за счет другой его структуры: в конфиге Gulp описывается сам процесс сборки, который должны пройти файлы. Пример конфиг-файла Gulp, собирающий CSS приведен ниже:

```

const gulp = require('gulp');
const sass = require('gulp-sass');
const autoprefixer = require('gulp-autoprefixer');
gulp.task(function sass() {
  return gulp.src('dev/*.scss')
    .pipe(sass())
    .pipe(autoprefixer())
    .pipe(gulp.dest('css/styles'));});
gulp.task('default', gulp.series('sass', function() {
  gulp.watch('dev/*.scss', gulp.series('sass'));});});

```

На данном этапе Gulp имеет 2844 плагинов (по состоянию на январь 2017 г.). Несмотря на гораздо меньший объем доступных плагинов (по сравнению с Grunt), возможно найти готовое решение для большинства типовых задач.

– Brunch. Основной целью разработчиков Brunch являлось создать инструмент, менее универсальный, чем Grunt, но конкретно ориентированный на задачи фронтенда, т.е. система без лишних настроек будет распознавать *.js как файл со скриптами, *.css как каскадные стили и т.д. Также как и в Gulp, в Brunch реализован принцип параллельного запуска задач. К достоинствам Brunch стоит отнести также компактный конфиг-файл, в несколько раз меньше Grunt и Gulp [3]:

```
exports.config =
  files:
  javascripts:
  joinTo:
  'javascripts/app.js': /^app/
  'javascripts/vendor.js': /^(bower_components|vendor)/
  stylesheets:
  joinTo: 'stylesheets/app.css'
  order:
  after: ['vendor/styles/helpers.css']
  templates:
  joinTo: 'javascripts/app.js'
```

Из примера видно, что на уровне конфига Brunch понимает, что требуется склеить файлы (joinTo), причем некоторые в определенном порядке (order). На данный момент для системы имеется небольшое количество плагинов (около 80), и развивается проект весьма медленно.

– Broccoli. Главной задачей при проектировании Broccoli, была реализация быстрых инкрементальных сборок, т.е. система сама понимает, какие файлы необходимо отслеживать, и будет пересобирать только те, которые требуется. Другая основная особенность Broccoli заключается в ее архитектуре: для описания входных и выходных данных используются не файлы, а деревья, т.е. директории с файлами, что позволяет гораздо проще описывать такие задачи, как, например, «многие-к-одному». Также разработчики Broccoli отказались от принципа параллельного запуска задач, и сделали выбор в пользу кеширования промежуточных результатов сборки вместо частичной пересборки требуемых файлов. Таким образом, когда требуется пересобрать проект или перезапустить отдельные плагины большая часть информации будет браться из кеша самих плагинов, что существенно экономит время [4].

К сожалению, пока расширений для Broccoli существует мало (около 20) и выполняют они только самые базовые задачи.

– Webpack. Стремительно набирающая популярность система сборки, определяемая как «сборщик модулей». Webpack основывается на двух принципах.

1. Модулем может быть не только js-файлы, а также и css- и html-файлы, а также изображения.
2. Модули должны загружаться тогда, когда это требуется: вместо того, чтобы генерировать все модули в один файл, Webpack позволяет делить код и асинхронно загружать необходимые части приложения.

Webpack имеет множество функций как для development-сборки, так и для production, в том числе предоставляет два интерфейса: webpack CLI и webpack-dev-server (Node.js сервер, предоставляющий возможности автоматической перезагрузки браузера, а также замены только что измененного модуля) [5].

Для расширения базовых возможностей в Webpack используются загрузчики и плагины.

Это лишь несколько наиболее популярных build-систем для фронтенда на платформе Node.js [6]. В настоящее время существует большой выбор готовых инструментов для решения задач автоматизации сборки фронтенда. Выбор конкретной build-системы стоит осуществлять по таким

факторам как функционал системы, а также его расширяемость, производительность, доступность документации, безусловно, опираясь при этом на специфику разрабатываемого проекта.

Литература

1. Тарасов А.Д., Алексанян Г.К., Хасанова М.Р. Автоматизация сборки JavaScript проектов с помощью Grunt // Новая наука: стратегии и векторы развития. – 2016. – № 118-2. – С. 211–213.
2. Grunt vs Gulp [Электронный ресурс]. – Режим доступа: <http://www.creative-seo.ru/blog/grunt-vs-gulp>, своб.
3. Brunch – ultra-fast HTML5 build tool [Электронный ресурс]. – Режим доступа: <http://brunch.io>, своб.
4. Broccoli: первый бета-релиз [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/216715/>, своб.
5. 10 особенностей Webpack [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/company/plarium/blog/309230/>, своб.
6. Node.Js Build Tools [Электронный ресурс]. – Режим доступа: <https://nodejs.libhunt.com/categories/478-build-tools>, своб.



Ставрова Владислава Сергеевна

Год рождения: 1995

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: vlstavrova@mail.ru

УДК 004.94

ИССЛЕДОВАНИЕ ТЕХНОЛОГИЙ ОПТИМИЗАЦИИ ТРЕХМЕРНЫХ МОДЕЛЕЙ

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 рассмотрено содержание файла трехмерной модели, рассмотрены технологии оптимизации трехмерных моделей путем редукции полигонов. Выявлены области применения трехмерных моделей и приведены примеры их использования, обозначены преимущества и недостатки использования оптимизированных трехмерных моделей.

Ключевые слова: трехмерное моделирование, оптимизация, полигоны, геометрия.

Оптимизация трехмерной модели подразумевает сокращение объемов информации, содержащейся в файле, с целью улучшения производительности приложения, в котором она используется. Потребность в оптимизации возникла в связи с распространенным использованием сложных трехмерных сцен для различных целей, а также появлением мобильных устройств и процессоров с низким уровнем производительности [1, 2].

Была поставлена **цель исследования**: провести анализ технологий оптимизации трехмерных моделей и их применения в разных сферах.

Для достижения цели необходимо было решить следующие задачи:

1. выявить основные технологии оптимизации;
2. обозначить основные области применения трехмерной графики;

3. выделить преимущества и недостатки оптимизированной модели.

Для того чтобы выявить технологии оптимизации, нужно иметь понятие о структуре файла трехмерной модели, и выделить те его части, для которых наиболее требуется сокращение объемов информации.

Данные, содержащиеся в файле трехмерной модели, разделяются на два класса: информация о геометрии объекта и информация о материалах. Информация о геометрии состоит из:

1. трехмерных координат вершин (X, Y, Z);
2. данных о гранях модели. В случае, если грань является треугольником, то она определяется тремя номерами ее вершин;
3. информации о нормалях, содержащихся в вершинах. Нормали являются векторами, используемыми при расчете освещения;
4. UV-развертки модели – двумерных координат, соответствующих трехмерным для каждой вершины.

Информация о материалах может включать в себя параметры для отражающей способности поверхности, цвета, текстур и т.д.

Поскольку информация о материалах занимает меньше памяти, чем информация о геометрии, вопрос оптимизации геометрии является более актуальным при работе с трехмерной графикой [3]. В этом случае задача сводится к уменьшению количества полигонов или треугольников. Это контролируется с начального этапа разработки модели.

Технологии оптимизации определяются из:

1. выбора программного пакета для работы с трехмерной графикой;
2. способов использования средств пакета.

Программные пакеты можно классифицировать как:

- использующиеся для полигонального моделирования. Наиболее распространенные редакторы – Autodesk Maya, Blender, Autodesk 3ds Max;
- использующиеся для интерактивного моделирования. Популярным примером является программа для скульптинга Zbrush.

Способы использования средств пакетов подразделяются на:

- удаление невидимых граней. При этом способе удаляются те полигоны, которые не будут видны при финальном просмотре сцены. Таким образом, объект может стать незамкнутым, в качестве примера у него может быть удалено невидимое дно. Этот способ используется в основном в пакетах для полигонального моделирования;
- редукция полигонов. Этот процесс сводится к упрощению сложной трехмерной модели до требуемого количественного значения полигонов. Редукция может осуществляться вручную (полигонально), но также существуют редакторы, позволяющие это делать автоматически (интерактивно);
- ручное создание низкополигональной модели. Чаще всего этот способ используется для экономии ресурсов в приложениях, работающих в реальном времени, либо в тех случаях, когда детализация не требуется. Качество модели восполняется при помощи использования материалов, включающих текстуры и карты. Основным достоинством ручного способа является высокое качество моделей, а недостатком является трудоемкость и низкая скорость моделирования. Этот способ применяется в полигональных редакторах.

При ручном создании низкополигональной модели используются такие способы упрощения трехмерной сетки, как:

1. соединение двух или более соседних вершин в одну. Такой способ применяется в случае, если несколько расположенных рядом вершин образуют нефункциональные полигоны. При этом могут быть удалены функциональные части модели, если ее внешний вид удовлетворяет требуемым эстетическим характеристикам;

- использование сглаженных нормалей вместо добавления дополнительных граней. Это позволяет таким объектам, как цилиндр, быть визуально воспринимаемыми при наименьшем количестве полигонов;
- замена полигонов треугольниками при моделировании или простое использование треугольников. В некоторых случаях это позволяет уменьшить суммарное количество полигонов [4].

Область применения трехмерной графики можно разделить на создание статичных изображений и использование в интерактивных приложениях. Для получения двумерного изображения из трехмерной сцены используются системы рендеринга, работу которых требуется оптимизировать в зависимости от оборудования, на котором производится рендеринг. Оптимизация необходима для сокращения временных затрат до получения финального результата. Известно, что увеличение одного из компонентов файла с трехмерной сценой, например, полигонов ведет к заметному падению скорости визуализации [1].

Наряду со статичными изображениями популярно использование интерактивной трехмерной графики для таких целей, как:

- визуализация игрового процесса, трехмерная графика позволяет пользователю чувствовать эффект виртуальной реальности;
- работа с наглядными учебными пособиями, когда вместо статичной картинки можно рассмотреть объект с разных сторон;
- представление архитектурного проекта, который планируется воплотить в будущем;
- выставочные инсталляции, наглядные и привлекающие посетителей.

Область применения интерактивной графики не ограничивается перечисленными пунктами, так как трехмерная графика является универсальным инструментом для визуализации любых объектов из мира и человеческой фантазии.

Очевидно, что основным преимуществом оптимизированной трехмерной модели является улучшенная производительность при ее визуализации. Недостатком является потеря качества и эстетических свойств модели. В этой связи при ее создании либо модификации учитывается, сколько ресурсов допустимо использовать при визуализации конкретной задачи, а также уменьшенное количество полигонов компенсируется дополнительными средствами, такими как использование текстур, карт, сглаженных нормалей и настроек материалов.

Примером способов, используемых для улучшения реалистичности/качества модели, является использование карт нормалей при создании материала модели. Карты нормалей – это цветное изображение, соответствующее модели по координатам UV-развертки, в котором красные, синие и зеленые цвета соответствуют координатам x , y и z вектора нормали к поверхности модели. Карта нормалей не может дать полное описание поверхности, а содержит лишь данные о направлении в каждой точке, и, таким образом, позволяет объекту смотреться рельефным [5].

Таким образом, в ходе исследования были выявлены технологии оптимизации трехмерных моделей путем уменьшения количества полигонов, рассмотрены области применения трехмерных моделей, а также обозначены основные преимущества и недостатки оптимизации.

Литература

- Котов Д.С. Технология упрощения геометрии на ранней стадии подготовки рендера // Алгоритмы, методы и системы обработки данных. – 2008. – № 13. – С. 87–92.
- Тимофеев А.В. Разработка и повышение производительности параллельной системы визуализации трехмерных сцен: автореф. дис. на соиск. учен. степ. канд. техн. наук. – СПб.: СПбГЭТУ «ЛЭТИ», 1997. – 16 с.

3. Посконин А., Зорин Д., Федюков М. Оптимизация анимированных полигональных моделей для web // Новые информационные технологии в автоматизированных системах. – 2012. – № 15. – С. 24–26.
4. Бояршинов А.Д., Ромин Е.А. Методы оптимизации высокополигональных моделей // Инновационные технологии: теория, инструменты, практика. – 2014. – № 1. – С. 301–310.
5. Кровопусков А.В., Рындин А.А. Сканирование карт нормалей поверхностей реальных объектов на основе сферического градиентного освещения // Вестник Воронежского государственного технического университета. – 2009. – № 11. – С. 177–179.



Сычева Анастасия Владимировна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: kir-dreamer@yandex.ru

УДК 004.514

ПСИХОЛОГИЯ ЮЗАБИЛИТИ

Сычева 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».

Проектирование интерфейсов сайтов, приложений и программ напрямую связано и испытывает влияние психологии пользовательского восприятия, его когнитивных процессов. Также происходит обратный процесс. Цепочка юзабилити – психология пользователя позволяет рассмотреть процессы, оказывающие влияние на проектирование интерфейсов, и обеспечить продуктивную работу человека с интерфейсом.

Ключевые слова: юзабилити, графический интерфейс пользователя, психология, пользовательское восприятие, когнитивистика, когнитивные процессы, дизайн, веб-дизайн.

Любое взаимодействие человека с окружающим миром строится на его восприятии окружающей действительности, его привычках и потребностях. Независимо от того, в какой среде находится пользователь, реальной или виртуальной, его действия подчинены его психологическим особенностям.

Юзабилити (англ. usability – дословно «возможность использования», «способность быть использованным», «полезность») – эргономическая характеристика того, насколько продукт может быть эффективно, экономично и с удовольствием использован определенными пользователями для достижения поставленных целей в заданном контексте использования (определение по стандарту Международной организации по стандартизации ISO 9241–11 «Руководство по юзабилити») [1]. Якоб Нильсен рассматривал пять составляющих юзабилити: обучаемость, эффективность, запоминаемость, ошибки, удовлетворенность [2].

В основе юзабилити лежит понимание пользовательской психологии. Нередко сайт заранее предлагает пользователям решение проблемы, о которой они только задумались. Психология юзабилити строится на привычках, культурных и традиционных особенностях, менталитете, жизненном опыте, а также на психологических трюках, которые позволяют пользователю испытывать удовлетворение от посещения сайта и работы с ним.

Практически сразу же после появления сети Интернет и создания первых сайтов разработчики пришли к выводу, что необходимы общие правила создания сайтов, чтобы элементы на странице воспринимались одинаково всеми людьми независимо от национальности, культуры и жизненного опыта. Работа над юзабилити включает в себя как интернационализацию, так и локализацию. Несмотря на то, что юзабилити приходит к интернациональному подходу, при его проектировании учитываются и локальные особенности человеческого восприятия [3]. При интернационализации сайтов и приложений разработчики переходят к унификации, когда схожие продукты выглядят одинаково.

Один из известных зарубежных исследователей психологии в вебе, Натали Нахай, считает, что психология действий в интернете схожа у многих людей [4]. Анализируя данные различных исследований, Н. Нахай пришла к выводу, что действиями пользователя можно манипулировать, если знать особенности психики. К примеру, пользователя могут в легкой форме «заставить» заполнить анкету (форму) или подписаться на обновления, если использовать эффект незавершенного действия. Результаты проведенных исследований выявили две группы пользователей: первая группа объединяет пользователей, которые предпочитают доводить дело до конца и, видя индикатор шага, они двигаются вперед, выполняя нужное действие, чтобы его завершить. Ко второй группе относятся пользователи, которые с трудом подключаются к выполнению действий, и такой индикатор поможет им завершить действие. Между тем этот эффект может быть использован не только как инструмент увеличения конверсии, но и как функция повышения удобства пользователя.

В. Завертайлов, генеральный директор студии интернет-решений «Сибирикс», в своей статье о юзабилити говорит о нескольких способах создания качественного продукта: «Самые лучшие новаторские интерфейсы вызывают к интуиции, рефлексам человека. Чуть менее простые для понимания базируются на способности человека самостоятельно находить взаимосвязи, анализировать» [5]. Оба этих принципа напрямую связаны с психологией и когнитивными способностями человека. Но при неточной формулировке или сложной схеме управления интерфейсом эти способы могут дать совершенно обратный эффект: пользователь не поймет, как работать с сайтом или приложением.

Пользователь нередко может ошибаться при работе с сайтом, который он видит впервые. Например, пользователь может пытаться нажать на несуществующие кнопки, теряться в меню и навигации на сайте. Зачастую это происходит от того, что ожидания пользователя не совпадают с реальным положением объектов на сайте. При работе с любым сайтом у пользователя «включаются» когнитивные функции: пользователь не только учится, как пользоваться сайтом, но и запоминает конкретные действия и функции [6]. Эти знания, сохраняемые человеческой памятью, становятся стереотипом поведения пользователя на сайте, повторяемым паттерном поведения. Память пользователя подсказывает, как должны вести себя те или иные элементы страницы, если они похожи на те, с которыми пользователь уже сталкивался ранее. Развитие когнитивной психологии способствовало созданию базовых правил оформления дизайна сайта, для того чтобы пользователь быстро и легко мог распознать однотипные элементы на разных страницах: например, ссылки всегда подчеркиваются и выделяются синим. Несоблюдение этих правил сильно снижает юзабилити сайта. Происходит психологический диссонанс, когда пользователь ожидает от элемента определенного внешнего вида или предсказуемой реакции, но не получает этого.

Важными характеристиками удовлетворенности пользователя при взаимодействии с сайтами являются время и скорость загрузки сайта и его отдельных страниц. Ожидание вызывает у пользователя беспокойство, ведь пользователь теряет свои ресурсы, в данном случае – время. В 2010 году поисковая система Bing проводила исследования [7] о зависимости удовлетворенности пользователя от скорости загрузки сайта и о влиянии этих критериев на доход компании, количество кликов. Результаты исследований показали, что если длительность загрузки сайта превышает две секунды, удовлетворенность пользователей снижается на 3,8%, доход с каждого пользователя сокращается на 4,3%, количество кликов уменьшается на 4,3%. На это влияет не

только нежелание пользователя тратить свое время, но и рассеивание пользовательского внимания в связи с ожиданием.

При работе с сайтом или приложением пользователь управляет своим вниманием. В зависимости от ситуации внимание пользователя может быть нескольких типов: от рассеянного внимания, когда пользователь воспринимает информацию в фоновом режиме, когда занят несколькими делами одновременно, до сфокусированного внимания, когда процесс полностью поглощает пользователя. На внимание пользователя и его восприятие оказывают влияние следующие факторы:

1. мотив пользователя;
2. цель и задачи, которые пользователь собирается решить;
3. имеющийся опыт и знания. Если человек прежде сталкивался со схожим интерфейсом, то ему будет проще работать и его внимание может быть менее сфокусированным. Если же пользователь впервые решает поставленную задачу новым для себя способом, то его внимание будет более сконцентрированным;
4. содержание сообщения. Внимание пользователя зависит от того, насколько понятен для него смысл сообщения, насколько удобно считывать информацию и каково ее визуальное представление;
5. тип пользовательского поведения на сайтах. По мнению Брайана и Джеффри Айзенберга, пользовательское поведение характеризуется четырьмя паттернами или типами: методический, спонтанный, гуманистический и конкурирующий тип [8]. В зависимости от ситуации и преследуемой цели поведение пользователя на сайте может быть разным, хотя в большинстве случаев пользователь придерживается одного типа поведения, проверенного длительным личным опытом.

Исследование пользовательской психологии при взаимодействии с интерактивными позволяет рассмотреть, на какие особенности пользовательского восприятия и психологии должна опираться разработка юзабилити, чтобы создать сайт или продукт, который не только отвечает требованиям технического задания, но и благоприятно воздействует на пользователя: вызывает доверие, экономит пользовательские ресурсы, быстро и эффективно решает пользовательскую задачу и оставляет положительный опыт после использования.

Литература

1. Большой психологический словарь / Под ред. Б.Г. Мещерякова, В.П. Зинченко. – М.: Прайм-ЕВРОЗНАК, 2003. – 672 с.
2. Нильсен Я. Веб-дизайн. – СПб.: Символ-Плюс, 2006. – 512 с.
3. Венда В.Ф., Венда В.Ю., Пашук Л.А. Инженерная психология в оценке и проектировании информационной техники («Юзабилити»): Российские приоритеты // Психологический журнал. – 2013. – Т. 34. – № 2. – С. 129–139.
4. Nahai N. Webs of Influence: The Psychology of Online Persuasion. – 2nd ed. – FT Press, 2017. – 264 p.
5. Завертайлов В. Юзабилити. Что победит – унификация или новаторство? [Электронный ресурс]. – Режим доступа: <http://blog.sibirix.ru/2014/01/28/who-wins/>, своб.
6. Ahlstrom U., Friedman-Berg F.J. Using eye movement activity as a correlate of cognitive workload // International Journal of Industrial Ergonomics. – 2006. – V. 36. – № 7. – P. 623–636.
7. Schurman E., Brutlag J. The User and Business Impact of Server Delays, Additional Bytes, and HTTP Chunking in Web Search [Электронный ресурс]. – Режим доступа: <http://conferences.oreilly.com/velocity/velocity2009/public/schedule/detail/8523>, своб.
8. Eisenberg B., Eisenberg J. Call to Action: secret formulas to improve online results. – Austen, Tex.: Wisard Academy Press, 2005. – 288 p.

**Таскин Никита Евгеньевич**

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4206Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: n_taskin@mail.ru

**Готская Ирина Борисовна**Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна,

д.педагог.н., профессор

e-mail: iringot@mail.ru

УДК 32.019.51

ИССЛЕДОВАНИЕ МЕТОДОВ И СРЕДСТВ ТЕСТИРОВАНИЯ WEB-ПРИЛОЖЕНИЙ**Таскин Н.Е.****Научный руководитель – д.педагог.н., профессор Готская И.Б.**

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

Проведено исследование методов и средств тестирования Web-приложений, предложен наиболее продуктивный стек технологий и метод тестирования, применение которых принесет высокий уровень качества Web-приложений.

Ключевые слова: тестирование, верификация, валидация, HtmlUnit, jWebUnit, Selenium WebDriver, white-box testing, black-box testing, grey box.

Сегодня вычислительные и коммуникационные системы являются неотъемлемой частью нашей повседневной жизни. Web-приложения соединяют различные компании, отделы внутри них, а также простых пользователей. Web-приложения очень динамичны, а их функциональные возможности непрерывно растут. Безостановочно возрастает потоковый трафик средств информации и запросов. Вследствие этого усложняются и системы такого рода. Очевидно, что для понимания анализа, разработки и управления подобными системами нужны количественные методы и модели, которые помогают оценить различные сценарии функционирования, исследовать структуру и состояние больших систем. Как и во всех системах, основанных на взаимодействии между клиентом и сервером, уязвимости Web-приложений обычно возникают из-за некорректной обработки запросов клиента и (или) недостаточной проверки входной информации со стороны разработчика. Формирование и решение данных проблем происходит на этапе тестирования программного обеспечения (ПО).

Этап тестирования помогает сделать процесс разработки ПО более прозрачным и управляемым для всех участников проекта. Эта фаза позволяет ясно увидеть и понять эволюцию проекта, проблемные места в процессе разработки, а также информацию для принятия оперативных решений о готовности продукта или его версии к продуктивной эксплуатации.

Существующие на данный момент методы и средства тестирования действуют в рамках формального процесса проверки разрабатываемого программного продукта, что не дает однозначно устранить все ошибки и установить корректность его функционирования.

Для того чтобы перейти к описанию методов и средств тестирования, прежде всего, необходимо дать определения основным терминам, которые являются базисом для понимания этой области.

Тестирование – это процесс анализа ПО, направленный на выявление отличий между его реально существующими и требуемыми свойствами (дефект) и на оценку свойств ПО [1, 2].

Качество ПО (Software Quality) – это совокупность характеристик ПО, относящихся к его способности удовлетворять установленные и предполагаемые потребности [3].

Верификация (verification) – это процесс оценки системы или ее компонентов с целью определения, удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа [1].

Валидация (validation) – это определение соответствия разрабатываемого ПО ожиданиям и потребностям пользователя, требованиям к системе [4].

Существуют методологии тестирования, применение которых зависит от того, имеет ли разработчик тестов доступ к исходному коду тестируемого ПО, либо тестирование выполняется через пользовательский интерфейс или прикладной программный интерфейс.

При тестировании методом белого ящика (white-box testing) (рис. 1, а) разработчик теста имеет доступ к исходному коду программ и может писать код, который связан с библиотеками тестируемого ПО.

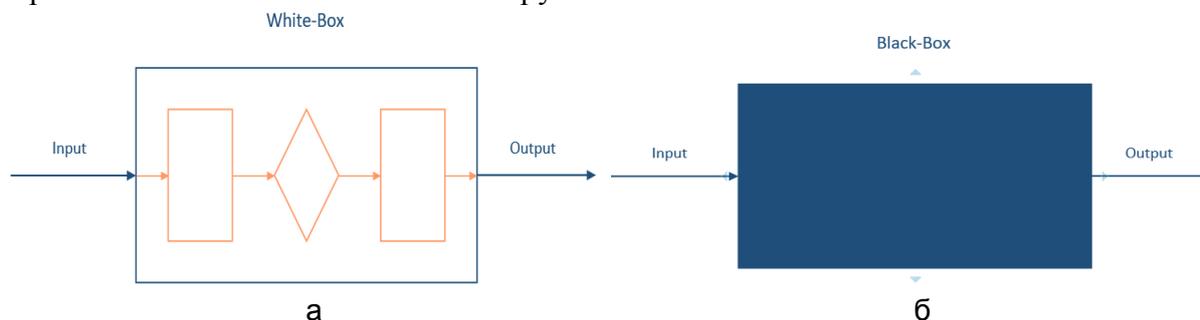


Рис. 1. Методы: белого ящика (а); черного ящика (б)

При тестировании методом черного ящика (black-box) (рис. 1, б) тестировщик имеет доступ к ПО только через пользовательский интерфейс, либо через внешние интерфейсы, позволяющие другому компьютеру, либо другому процессу подключиться к системе для тестирования.

Для того чтобы применять технологию или средство тестирования, прежде всего, нужно определиться с типом платформы тестируемого программного продукта и видом самого подхода к тестированию.

В Web-приложениях основным видом тестирования является функциональный, что приводит к применению определенных технологий тестирования, а именно:

- HtmlUnit;
- jWebUnit;
- Selenium WebDriver.

HtmlUnit. Технология HtmlUnit эмулирует поведение браузера через HTTP-запросы. Для большинства HTML-элементов предоставляется соответствующий объект с набором свойств и методов, через которые удобно анализировать и управлять HTML-документом.

JWebUnit. Технология jWebUnit обеспечивает высокий уровень Java API для навигации веб-приложения в сочетании с набором утверждений для проверки правильности приложения. Основным объектом при написании тестов является WebTester – абстракция веб-браузера. WebTester имеет набор методов, с помощью которых можно получить доступ к содержимому документа и управлять его состоянием.

Selenium WebDriver. Данная программная библиотека не имеет пользовательского интерфейса, она же является драйвером для браузера и через определенные команды позволяет разным другим программам взаимодействовать с ним, управлять его поведением и получать от него требуемые данные для прохождения теста.

В рамках работы были проанализированы описанные методы и технологии для тестирования программных продуктов, но в приведенном списке есть как положительные стороны, так и отрицательные.

В HtmlUnit используется движок rhino, из-за чего происходит неполноценная поддержка JavaScript, а тестировать DHTML и AJAX с его помощью практически невозможно.

Основным минусом в технологии Selenium WebDriver является то, что для выполнения тестов приходится запускать WebDriver как отдельный процессор. Также к минусам можно отнести и отсутствие встроенного механизма записи сообщений во время «прогона» теста.

Архитектура программного продукта jWebUnit позволяет подключать другие драйвера для выполнения web-тестов, а именно HtmlUnit и Selenium WebDriver, это позволяет использовать положительные стороны одной технологии над недостатками другой.

В методе белого ящика (white-box testing) создание тестов происходит на основе структуры самой системы. Оценка степени соответствия требований зависит от знания внутренних данных системы, что является весьма трудоемким и не всегда эффективным с точки зрения бизнес-логики создаваемого приложения.

Тестирование методом черного ящика (black-box testing) происходит на основе требований, описанных в техническом задании и функциональной спецификации, что не обязывает тестировщика знать о внутренней структуре тестируемой системы, но может привести к ошибкам на уровне кода.

Тестирование методом серого ящика (gray-box testing) (рис. 2) применяется в том случае, когда тестировщик, который разрабатывает тест-кейсы, имеет информацию о внутренней структуре проектируемого приложения и деталях реализации, что является наиболее эффективным методом для решения задач по тестированию, а в итоге может дать наиболее высокий уровень качества программного продукта.

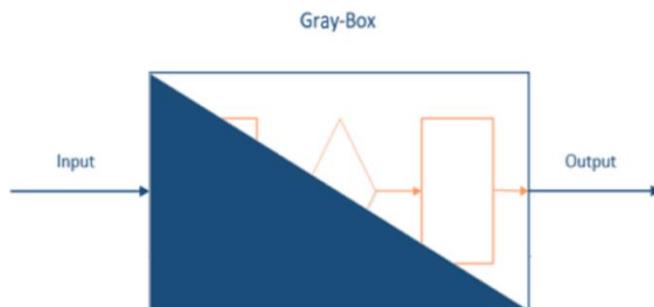


Рис. 2. Метод серого ящика

Литература

1. Guide to the Software Engineering Body of Knowledge [Электронный ресурс]. – Режим доступа: <http://www.math.unipd.it/~tullio/IS-1/2007/Appfondimenti/SWEBOK.pdf>, своб.

2. Standard glossary of terms used in Software Testing [Электронный ресурс]. – Режим доступа:
http://www.rstqb.org/fileadmin/user_upload/redaktion/rstqb_ru/downloads/ISTQB_Glossary_English_v2.3.pdf, своб.
3. ISO [Электронный ресурс]. – Режим доступа:
http://www.iso.org/iso/catalogue_detail.htm?csnumber=20115, своб.
4. TestingStandards [Электронный ресурс]. – Режим доступа:
http://www.testingstandards.co.uk/bs_7925-1.htm, своб.



Тодоровская Яна Васильевна

Год рождения: 1995

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: correct.V@yandex.ru

УДК 004.457

ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ OPEN SOURCE ПРОДУКТОВ ДЛЯ ФОРМИРОВАНИЯ АЙДЕНТИКИ КОМПАНИИ

Тодоровская Я.В.

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 было проведено исследование в области использования Open Source продуктов. Рассмотрены аналоги платных программ, таких как: Adobe Photoshop, Adobe Illustrator и CorelDRAW. Обоснована актуальность исследования данного направления.

Ключевые слова: айдендика, Open Source, Adobe Illustrator, Adobe Photoshop, CorelDRAW.

Айдендика (ее еще называют «фирменным» или «корпоративным стилем», «brand ID» или «corporate ID») – это визуальная составляющая бренда, призванная повысить его узнаваемость и создать впечатление целостности [1]. Айдендика включает в себя все, что относится к визуализации бренда. Актуальность данной темы заключается в том, что на сегодняшний день корпоративный стиль повышает доверие клиентов и партнеров, способствует продвижению общего имиджа и репутации компании на рынке. Также ведет к оптимизации целостности восприятия у клиентов, что позитивно складывается на росте известности компании и сокращает расходы на рекламную кампанию.

Для формирования айдентики компании используются профессиональные программы такие как Adobe Photoshop, CorelDRAW, Adobe Illustrator, они являются платными, тем самым не могут быть доступными всем пользователям.

Айдендика рассматривается как совокупность визуальных элементов, которые используются в различных приложениях для продвижения имиджа организации. Изначально этот термин был синонимом всех элементов, связанных с корпоративной деятельностью организации – документации, логотипов и визуальных элементов, а сейчас айдендика и корпоративная стратегия оказались тесно связаны [2]. Некоторые специалисты в области маркетинга считают, что айдендика выступает в качестве основного элемента существования организации, которая включает в себя свою историю, философию, технологию, людей, их

этнические и культурные ценности. Айдентика также помогает определить положение организации на рынке и возможности в конкурентной среде. Бренды всегда создают и изменяют визуальный образ, чтобы закрепить свою позицию в обществе, которое ориентировано на новые технологические инновации и современные средства связи, путешествия и развлечения. Образ бренда может быть сделан хорошо или плохо, и успех компании часто зависит от эффективности визуальных материалов, которые его представляют [3].

Айдентика компании разрабатывается как в растровых, так и в векторных редакторах, например, для создания качественного логотипа используются профессиональные платные редакторы, такие как CorelDRAW, Adobe Illustrator. Однако существуют бесплатные аналоги.

Перейдем к Open Source. Существуют следующие виды программ:

1. GIMP – это мощное бесплатное, кроссплатформенное приложение для работы с растровой и векторной графикой. Программа используется для обработки цифровых изображений и фото с целью создания логотипов, рисунков, изменения их размеров, манипуляции цветами, комбинирования, конвертации.

GIMP позволяет одновременно работать с несколькими файлами. В нем имеется специальный менеджер памяти, обеспечивающий обработку множества изображений без перегрузки компьютера.

Редактор работает и со слоями. Данная функция позволяет получать изображения из множества отдельных рисунков. Таким образом, каждый проект представляет собой совокупность слоев, частные случаи – изображения, состоящие из одного слоя.

Зачастую работа с картинками предусматривает выделение части рисунка для его дальнейшего редактирования. Инструмент выделения предназначен именно для этого. При выделении часть изображения разграничивается пунктирной линией. Доступно выделение произвольной области рисунка, при этом сохраняется высокая точность контура.

Иногда редактирование сопровождается ошибками, поэтому в программе предусмотрена возможность отмены действий. Вернуться в исходное положение пользователь сможет когда угодно, поскольку утилита сохраняет последовательность действий. Однако «история» занимает много памяти, следовательно, возможности отмены ограничены несколькими шагами. Во многом данная функция зависит от того, сколько памяти занимает то или иное действие.

Практически все функции редактора являются встроенными. Стоит отметить, что GIMP позволяет применять внешние инструменты – плагины, которые пользователь задействует в специфических ситуациях. Большинство плагинов поставляется с программой, но можно задействовать и сторонние, созданные другими разработчиками. Приложение может быть использовано как обычное средство для рисования или в качестве редактора для профессиональной ретуши цифровых изображений высокого класса.

GIMP позволяет выполнять пакетную обработку графики, в нем предусмотрена система публикации. Также пользователю доступен скриптинг – автоматизация простых и сложных задач [4].

2. Gravit – это программа для дизайна графических интерфейсов, которая работает в браузере, не требует установки и настроек. Главным плюсом является независимость от платформ, в ней неважно, какую операционную систему (Windows, Mac или Linux) вы используете. Недостатком является хранение всех дизайнов на сервере Gravit. Без доступа к интернету вы не сможете их использовать.

После регистрации на сайте пользователь получает собственное пространство, где хранятся его работы. Сами работы можно сделать публичными, доступными другим

пользователям, пометить как не завершенные, нуждаются в проверке, утвержденные и тому подобное [4].

В редакторе пользователь имеет уже привычный набор инструментов: указатель, перо, лассо, слои, нож, графические примитивы, линии, кривые Безье, группировку, слияние, кадрирование, преобразование, выравнивание и т.д. Реализован стандартный набор настроек для работы со шрифтами. Есть даже несколько с кириллицей. Существует импорт из SVG и PS/EPS форматов, и экспорт в SVG и растровые форматы PNG и JPG.

Для работы предлагается современный набор размеров полотна, активно формируется сообщество пользователей, наполняется библиотека готовых работ и клипарта (market).

3. Inkscape – векторный графический редактор – один из самых достойных бесплатных аналогов CorelDRAW и Adobe Illustrator. Удобен для создания как художественных, так и технических иллюстраций (вплоть до использования в качестве системы автоматизированного проектирования (САПР) общего назначения, чему также способствует легкость обмена чертежами). Это стало возможным во многом благодаря открытому формату SVG, развиваемому консорциумом W3C. Формат SVG позволяет создавать иллюстрации различного типа, в том числе анимированные. Поскольку SVG основан на языке разметки XML, к нему можно писать расширения, Приложение имеет стандартный интерфейс: меню, панели инструментов, палитра, не считая множества других диалогов.

В Inkscape практически полный функционал растрового редактора. Например, программа поддерживает различные виды смешивания.

Кроме рисования геометрических примитивов, Inkscape поддерживает также добавление текста с различными эффектами, в том числе написание текста по кривым линиям. В состав приложения входит огромное количество фильтров [4].

Целевое использование:

1. иллюстрации для офисных циркуляров, презентаций, создание логотипов, визиток, плакатов;
2. технические иллюстрации (схемы, графики и пр.);
3. векторная графика для высококачественной печати (с предварительным импортом svg в scribus);
4. веб-графика – от баннеров до макетов сайтов, пиктограммы для приложений и кнопок сайтов, графика для игр.

Таким образом, для формирования айдентики компании можно использовать различные Open Source продукты, однако, приобретение платных, лицензионных версий Adobe Photoshop, CorelDRAW, Adobe Illustrator позволят пользователю выполнить качественно поставленную задачу.

Литература

1. Кумова М. Айдентика. – М.: КАК Проект, Grey Matter, 2014. – 512 с.
2. Айдентика – это. Фирменная айдентика [Электронный ресурс]. – Режим доступа: <http://fb.ru/article/158649/aydentika---eto-firmennaya-aydentika>, своб.
3. Годин А.М. Брендинг: учебное пособие. – М.: Дашков и К, 2013.– 183 с.
4. Обзор трех популярных графических редакторов [Электронный ресурс]. – Режим доступа: <https://studylinux.ru/obzor-treh-populyarnyh-graficheskikh-redaktorov-linux.html>, своб.



Топорков Павел Сергеевич

Год рождения: 1980

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: Labour_80@mail.ru

УДК 007.51

СОВРЕМЕННЫЕ ВОПРОСЫ ИНТЕГРАЦИИ CAD И PLM

Топорков П.С.

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

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

Работа посвящена вопросам интеграции CAD и PLM. Рассмотрена концепция единого информационного пространства проектно-производственного предприятия. Обозначены современные тренды. Сформулированы основные требования к интеграции различных автоматизированных систем.

Ключевые слова: CAD, PLM, интеграция, проектирование, САПР, ЖЦИ.

Актуальность проблемы обусловлена определяющей ролью информационных технологий в формировании рабочего процесса современного промышленного и (или) проектировочного предприятия. Внедрение электронного документооборота, единой базы изготавливаемых изделий, покупных материалов и компонентов, организация совместной работы различных специалистов над одним проектом, в том числе и в реальном времени, возможность использования готовых наработок, трехмерных моделей, чертежей, результатов визуализации, справочных данных, спецификаций и иных технических документов в общем, цельном, интерактивном, информационном пространстве представляют колоссальные возможности по увеличению оперативности принятия решений, манипуляции частными процессами и задачами, сокращению сроков выполняемых проектов и, как следствие, увеличению экономической эффективности. В то же время оно немислимо без корректной передачи данных между различными системами и накладывает определенные ограничения на регламент представления и хранения информации как в каждом конкретном отделе/подразделении, так и для каждого конкретного сотрудника/специалиста, участвующего в процессе. Передача данных обязательно должна происходить по определенным единообразным правилам, обеспечение выполнения которых является приоритетной задачей на самом начальном этапе внедрения программного обеспечения (ПО). Цена ошибки в данном случае может быть непозволительно велика – вплоть до необходимости вносить изменения в проект уже на последнем этапе, когда он должен быть готов к сдаче, а подчас и производить неизбежную принципиальную переработку «с нуля» многих его разделов. По временным и энергетическим затратам это может быть сопоставимо с полным прохождением всех этапов проектирования заново, т.е. с необходимостью начинать весь проект сначала.

На более ранних исторических этапах системы автоматизированного проектирования (САПР, они же – Computer aided design, CAD) функционировали независимо друг от друга и различных иных систем. Во главу угла ставился непосредственно процесс проектирования. Несколько позже появились системы

управления жизненным циклом предприятия (Product Lifecycle Management, PLM). С точки зрения разных видных специалистов в области САПР и PLM в истории САПР, управления данными и совместной разработки тема интеграции CAD/PLM является наиболее интересной [1].

В строгом смысле система автоматизации проектных работ (САПР) – это система, обеспечивающая выполнение функций проектирования и подразумевающая частичную или полную автоматизацию типовых операций [2]. Под САПР подразумевается не только ПО, но и вся совокупность технических средств и организационных мероприятий, направленных на обеспечение процесса проектной (и не только) деятельности предприятия.

Однако то же самое можно утверждать и по поводу PLM. Согласно одному из самых распространенных определений PLM-система – прикладное программное обеспечение, предназначенное для управления жизненным циклом изделия [3]. Жизненный цикл изделия (ЖЦИ) содержит в себе несколько этапов от появления идеи нового продукта до момента утилизации по истечении срока его использования [2].

Упорядоченный перечень всех видов деятельности и операций, производимых при осуществлении проекта и изготовлении изделий, называется производственным циклом. Решение поставленных задач на современных предприятиях, выпускающих сложную наукоемкую продукцию, не представляется возможным без полноценного внедрения автоматизированных систем (АС). По причине специфичности задач, решаемых на различных этапах жизненного цикла изделий, существует несколько различных подвидов АС.

В русском языке термин САПР подразумевает интегрированную CAD/CAM/CAE/PDM-систему. По функциональной нагрузке различают подсистемы САПР, поддерживающие разные аспекты проектно-производственной деятельности.

CAD – средства автоматизированного проектирования, термин охватывает средства САПР, предназначенные для автоматизации двумерного и (или) трехмерного геометрического проектирования, создания конструкторской и (или) технологической документации, САПР общего назначения. Иногда для определения данного класса средств САПР используется также термин CADD (computer-aided design and drafting) – автоматизированное проектирование и создание чертежей.

CAM – средства технологической подготовки производства изделий, обеспечивающие автоматизацию программирования и контроля оборудования с числовым программным управлением (ЧПУ) или гибких автоматизированных производственных систем (ГАПС).

CAE – средства автоматизации решения инженерных задач, анализа и симуляции физических процессов, осуществления моделирования, верификации и оптимизации разрабатываемого продукта.

Системы для работы с данными об изделии (PDM-системы) используются на всех этапах проектирования, предоставляя возможность работы в режиме коллективного проектирования, автоматизации части задач управления.

Информация об изделии – это совокупность данных, порождаемых и используемых на всех этапах жизненного цикла (ЖЦ).

С учетом всего вышеизложенного, уместнее говорить не об интеграции САПР или их отдельных подсистем с какими-либо внешними дополняющими системами, а о создании единого информационного пространства, интегрированной системы автоматизации (ИСА), закрывающей гораздо более широкий спектр вопросов, чем САПР или даже PLM.

Необходимость интеграции различных систем выдвигает ряд требований к ним, в том числе: поддержка парадигмы твердотельного; перенос ассоциативных связей на все уровни проекта; обеспечение горизонтальной и вертикальной интеграции и

сбалансированности модулей в рамках единой системы; наличие средств поддержки параллельного проектирования и методов коллективной работы [2]; возможность постоянной синхронизации информации, находящейся под управлением CAD и PLM; возможность создания стандартных компонентов и объектов модели как в САПР, так и в PLM; охват интеграцией как можно более широкого диапазона этапов ЖЦИ; поддержка обмена атрибутами между CAD и PLM; возможность работы с уникальными идентификаторами объектов как в CAD, так и в PLM; поддержка контроля доступа к объектам CAD, основанного на принципах безопасности и степени готовности информации; возможность автоматической генерации спецификаций, чертежей в PLM как результата публикации объектов модели и чертежей в CAD; возможность переноса технологического дерева изготовления изделия из CAD в PLM и наоборот [4]. Решающую роль для обеспечения открытости САПР, возможности ее интеграции с другими АС играют интерфейсы, представляемые реализованными в системе стандартами межпрограммных обменов [2].

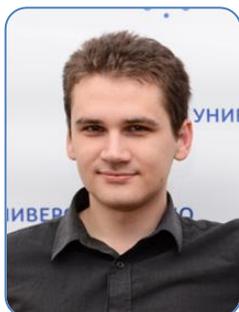
Еще более высоким уровнем абстракции являются CALS-технологии (Continuous Acquisition and Life cycle Support – непрерывная информационная поддержка поставок и ЖЦИ) – современный подход к проектированию и производству высокотехнологичной и наукоемкой продукции, состоящий в использовании вычислительной техники и современных информационных технологий на всех стадиях ЖЦИ.

ИПИ (информационная поддержка процессов жизненного цикла изделий) – русскоязычный аналог понятия CALS. Стратегия CALS объединяет в себе следующие основные принципы: использование новейших информационных технологий; единообразие представления и интерпретации данных в процессах информационного взаимодействия между автоматизированными системами и их составными частями; доступность информации для всех участников ЖЦИ в произвольное время и в любом месте; разработка и модификация бизнес-процессов; поддержка совмещенного (параллельного) проектирования изделий; унификация и стандартизация средств совместного использования АС и электронного обмена данными; интеграция образования и производства, связь эффективности производства с качеством подготовки и переподготовки специалистов [2].

Резюмируя, можно сделать следующий вывод. На данном этапе развития технологий автоматизации процессов производства и проектирования основным трендом является интеграция – объединение всех используемых автоматизированных систем в единое информационное пространство. Причем речь идет не только об интеграции программных модулей, но и организационных процессов, и участвующих специалистов в единую органичную бесперебойно действующую систему, т.е. о формировании концепции, системы правил и стандартов с использованием новейших достижений в области информационных технологий и телекоммуникации.

Литература

1. Шиловский О. Как заново разработать интеграцию CAD и PDM? [Электронный ресурс]. – Режим доступа: http://isicad.ru/ru/articles.php?article_num=16968, своб.
2. Муленко В.В. Компьютерные технологии и автоматизированные системы в машиностроении. – М.: РГУ нефти и газа им. И.М. Губкина, 2015. – 73 с.
3. Травин А. Teamcenter. Интеграция с CAD-системами [Электронный ресурс]. – Режим доступа: http://www.cadmaster.ru/magazin/articles/cm_75_09.html#hcq=IXJ7H1q, своб.
4. Томан М., Перез Р. Интеграция БД ориентированных САПР с системами PLM в кораблестроении [Электронный ресурс]. – Режим доступа: remmag.ru/upload_data/files/2-2014/SENER, своб.



Туркунов Александр Алексеевич

Год рождения: 1991

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4106

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: snowfayu@gmail.com

УДК 004.92

ПЕРСПЕКТИВЫ РАЗВИТИЯ ТЕХНОЛОГИИ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ В ТЕХНИЧЕСКИХ СИСТЕМАХ

Туркунов А.А.

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

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

В работе проведен анализ развития технологий виртуальной реальности в технических системах, их дальнейшие перспективы развития, рассмотрены их основные достоинства и недостатки.

Ключевые слова: технологии, виртуальная реальность, новинки, рынок, перспективы.

История уже знает несколько провальных попыток продвижения данной технологии. Многие окрестили идею виртуальной реальности в том виде, что ее показывают сейчас – обреченной на провал и являющейся неперспективным проектом. И не без причины.

Цена. На рынке уже несколько очков виртуальной реальности (VR) от разных производителей (окулус рифт, HTS, project morpheus/Sony) и, несмотря на то, что компании позиционируют свою продукцию как доступную для широкого круга пользователей, их цена варьируется от 350\$ до 800\$. И это цена за не самостоятельный девайс. Даже для Американского и Европейского рынка цена не маленькая.

Характеристики. Окулус Рифт был представлен в 2012 году на E3 (сокр. Electronic Entertainment Expo – выставка электронных развлечений), и с тех пор он становился легче, меньше, но все равно его конечный вариант выглядит громостким, не практичным и просто не эргономичным. Помимо физической усталости очки могут вызывать различные проблемы со зрением и тошноту. Из-за этих проблем многочасовая работа в таком девайсе становится затруднительной. Подумайте, какая это физиологическая нагрузка и нагрузка на мозг. Все это приводит к мысли о том, что окулус в этом плане технологии недалеко ушел от своих прототипов, вроде шлема виртуальной реальности VFX-1, разработанного американской фирмой Forte Technologies, в теперь уже далеком 1994 году.

Главная проблема. На данный момент главное, что мешает развитию технологий VR в области развлечений, – это отсутствие у разработчиков необходимых ресурсов и клиентской базы. Пользователи, в свою очередь, не до конца доверяют разработчикам программного обеспечения для VR из-за того, что нет качественной продукции, и далеко не у всех есть возможность потратиться на подобный не самостоятельный девайс. В итоге получается своеобразный замкнутый круг – так называемая Уловка-22 (крайняя форма взаимоисключающих параграфов, вплоть до полного и тотального взаимоисключения) [1–10].

Все это превращает шлем ВР не в новую ступень технологий, а в аттракцион. Аттракцион для домашнего одноразового пользования на подобии Kinect и Playstation move.

Но несмотря на все это, последнее время данные технологии получили большое развитие. 2016 год вообще называют годом виртуальной реальности. На сегодняшний день «Большой тройкой» среди шлемов ВР называют Oculus, Sony, HTC. Oculus в течение прошлого года был реализован в количестве 350 тыс, HTC Vive – 450 тысяч, а Sony PlayStation VR – 750 тысяч. Эти устройства разработчикам не удалось реализовать миллионными тиражами, но предполагается, что в будущем году это может получиться. Правда в отношении Oculus этим прогнозам скорей всего не суждено сбыться. В начале февраля 2017 г. компания ZeniMax отсудила у Oculus Rift 500 млн долл. за нарушение NDA, одним из основателей компании, а также авторских прав. Теперь же ZeniMax обратились в суд с требованием запрета «использования, маркетинга, продажи, дистрибуции, изменения сервиса, копирования или предложения для продажи или лицензирования любых продуктов, которые полностью или отчасти используют в каком-либо виде для любой задачи какие-либо материалы, защищенные копирайтом». Другими словами, сами очки ВР не исчезнут с рынка, но есть вероятность потерять софт, стоящий за ними, и это может нанести серьезный удар компании. Если запрет будет принят, то Facebook придется серьезно задуматься о сделке с ZeniMax.

С развитием и популяризацией цена продолжит падать, что положительно скажется продажах и немного ослабит замкнутый круг Разработчик-Клиент-Разработчик и как следствие, появление проектов.

Характеристики продолжают улучшаться. Как автор работы уже сказал ранее: «он становится легче, меньше», и возможно последующие прототипы смогут приблизиться к приемлемым показателям. Причиной тому послужило развитие таких технологий, как датчики положения с высокой чувствительностью, экраны с высокой частотой развертки и другие. Так например с увеличением частоты кадров, исчез эффект укачивания и мигрени. Автор на своем опыте смог в этом убедиться пока в течении получаса использовал HTS. Никаких негативных ощущений не было. Люди уже ставят эксперименты, проводя в очках сутки и более.

Также хотелось бы заметить, что проекты виртуальной и дополненной реальности не ограничиваются индустрией развлечений, и могут не только создавать концептуально новые рынки, но и расширять уже имеющиеся:

- прямые трансляции с эффектом присутствия. Это – возможность ощутить эффект присутствия на различных мероприятиях – таким образом, можно избежать покупки дорогостоящих билетов и не тратить время на дорогу. Радио и телевидение служат тем же целям, но технологии ВР открывают абсолютно новые возможности и границы ощущений;
- индустрия кино. Зрители смогут полностью окунуться в фильм, а не смотреть его со стороны;
- продажа недвижимости/продажи. Технологии ВР откроют новые возможности для покупателей в сфере продажи недвижимости и интернет-покупок. Некоторые компании уже сейчас разрабатывают функции просмотра жилья и мебели в виртуальном пространстве;
- образование. Применение технологий ВР в данной сфере предоставит учащимся интерактивные и наглядные схемы и макеты, которые упростят процесс обучения по таким направлениям как физика, химия, биология, история, геометрия и математика. Google бесплатно продвигает в некоторых школах свой проект Cardboard, который к началу 2016 года подготовил более 100 учебных программ. Кроме того, проектами виртуальной и дополненной реальности интересуются многие медицинские образовательные учреждения;
- здравоохранение. Подспорье в работе хирургов, лечение различных фобий и психических расстройств;
- проектирование. Применение технологий виртуальной и дополненной реальности даст уже имеющимся компьютерным технологиям проектирования (автоматизация производственных процессов, система автоматизированного проектирования) новые инструменты и возможности.

Появится возможность проводить предварительные испытания выпускаемой продукции в ВР, благодаря чему снизятся риски и конечная стоимость детали;

- военная промышленность. Обучение пехоты, медиков и летчиков в условиях максимально приближенных к реальности, но без какой либо опаски для жизни и здоровья.

В каждом из этих направлений технологии ВР, даже в том виде, что мы имеем сейчас, имеют гигантский потенциал. Нужно только время.

Кроме того, думать нужно масштабнее. Если бы кто-то еще пару лет назад сказал, что мы с вами сможем приобрести устройства для ВР в домашнее пользование, ему бы никогда не поверили. Потому даже вообразить себе трудно, что мы можем увидеть через десятилетие.

ВР – это все-таки ступенька нашей с вами эволюции. Возможно это и не наше будущее, но тот первый шаг, который в дальнейшем изменит мир.

Литература

1. 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. 24–26.
2. Антонова О.А., Соловьев С.В. Теория и практика виртуальной реальности. Логико-философский анализ. – СПб.: Изд-во Санкт-Петербургского ун-та, 2008. – 168 с.
3. Россохин А.В., Измагурова В.Л. Виртуальное счастье или виртуальная зависимость. Личность в измененных состояниях сознания. – М.: Смысл, 2004. – С. 516–523.
4. Войскунский А.Е., Смылова О.В. Психология применения систем виртуальной реальности // Интернет и современное общество. Труды IX Всероссийской объединенной конференции. – 2006. – С. 52–54.
5. Игнатъев М.Б., Никитина А.В., Войскунский А.Е. Архитектура виртуальных миров. – СПб.: Изд-во ГУАП, 2009. – 287 с.
6. Клайн Э. Первому игроку приготовиться. – Изд-ва: АСТ, Neoclassic, 2013. – 480 с.
7. Маньковская Н.Б., Бычков В.В. Виртуальность в пространствах современного искусства // Сб. научно-популярных знаний победителей конкурса РФФИ. – 2007. – № 10. – С. 374–380.
8. Дацюк С. Теория виртуальности [Электронный ресурс]. – Режим доступа: http://www.uis.kiev.ua/xyz/tv/tvbook.htm#_Тoc215117724, своб.
9. Google I/O 2016 в подробностях: перспективы и технологии [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/company/google/blog/301504/>, своб.
10. 9 сфер применения виртуальной реальности: размеры рынка и перспективы [Электронный ресурс]. – Режим доступа: <https://vc.ru/p/vr-use>, своб.



Шуклин Дмитрий Анатольевич

Год рождения: 1979

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, к.педагог.н.

e-mail: kpd@limtu.ru



Тхорёва Лидия Алексеевна

Год рождения: 1992

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4205

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: tkhoreva_li@rambler.ru

УДК 32.019.51

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ОНЛАЙН-СЕРВИСОВ ДЛЯ ПРОДВИЖЕНИЯ ГРУПП ВКОНТАКТЕ

Тхорёва Л.А.

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

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

Работа посвящена сравнительному анализу онлайн-сервисов для продвижения групп «ВКонтакте». В работе описаны популярные онлайн-сервисы для продвижения сообществ в социальной сети «ВКонтакте», и предоставляется их краткая характеристика. Констатируется, что наиболее эффективное продвижение достигается за счет правильной постановки целей в развитии группы, определения целевой аудитории, а также использования онлайн-сервисов для продвижения.

Ключевые слова: социальные сети, этапы продвижения, продвижение сообществ, продвижение проектов в «ВКонтакте», сервисы для продвижения, раскрутка «ВКонтакте».

Согласно статистике за месяц социальную сеть «ВКонтакте» посещает порядка 90 млн человек. Сотни тысяч людей, каждый из которых имеют свои вкусы, увлечения и жизненные взгляды, представляют для любого бизнеса огромную базу потенциальных клиентов. Большая часть мировых брендов на сегодняшний день имеют официальные сайты, страницы и аккаунты в соцсетях. Данная тенденция набирает все больше оборотов в последние годы, и даже небольшие компании, которые не обладают широкой популярностью, создают социальные сообщества, чтобы стать ближе к своим клиентам. Грамотный подход к рекламированию своей деятельности в соцсетях помогает предпринимателям расширить свой бизнес, а специальные онлайн-сервисы для продвижения сообществ являются залогом успешного сбора статистики и анализа для качественного продвижения.

В работе рассмотрены онлайн-сервисы для продвижения групп «ВКонтакте», а также предложены наиболее оптимальные сервисы для работы с сообществом.

Онлайн-сервисы, пользующиеся большой известностью, можно разделить на сервисы-анализаторы, сервисы-автопостинги и сервисы, которые занимаются таргетингом.

Рассмотрим сервисы-анализаторы:

1. JagaJam.com. Очень популярный сервис среди интернет-маркетологов из-за того, что собирает максимум статистических данных. Сервис легкий и понятный, при использовании выходят всплывающие окна с подсказками, а также есть справка, где можно найти ответы на интересующие вопросы. JagaJam – платное приложение, и используется в основном юридическими лицами. Позволяет анализировать несколько сообществ и их пересечения. Отдельным плюсом является то, что можно выгружать отчет в формате Excel и автоматически отправлять его на электронную почту [1].
2. Popsters.ru. Цель данного сервиса – проанализировать, чем увлекается аудитория и найти самый интересный контент. Собрать данные можно по любому сообществу: узнать коэффициент вовлеченности, количество записей на стене, какая аудитория у сообщества. Главной ценностью этого приложения является добавление ленты записей за

определенный период, сортировка по комментариям, лайкам, дате, коэффициенту эффективности [1]. Сервис платный.

3. **Borgi.ru**. Данный сервис является бесплатным анализатором сообществ. Он предоставляет полную статистику по сообществам «ВКонтакте», анализирует подписчиков по полу и возрасту. В этом приложении можно найти своих конкурентов, сравнить цены на рекламу на биржах, и найти площадки для размещения рекламируемой информации.

Перед выбором сервиса-анализатора для продвижения сообщества «ВКонтакте» сразу стоит определиться, есть у вас бюджет или нет. Если нет, то на помощь придет **Borgi.ru** – сервис с бесплатным доступом. Среди платных сервисов многие интернет-маркетологи выбирают **JaJaJam.com** из-за большого количества предоставляемых статистических данных.

Сервисы-автопостинги:

1. **Postio.ru**. Данный сервис автоматически наполняет сообщества и работает, как большинство автопостингов – запускает в определенные часы нужные посты по заданным тематикам. Платный, можно использовать для нескольких социальных сетей.
2. **SMM-box**. Сервис занимается автопостингом «ВКонтакте» и самостоятельно ищет контент. Можно поставить отбор по тематике групп, контенту. По итогу анализирует результаты проделанной работы. Плюсом является возможность бесплатной работы в данном приложении в течение 14 дней.
3. **Sociate** – биржа рекламы «ВКонтакте». Удобный и качественный сервис, который автоматически запускает рекламу, посты в сообществах, а также подбирает лучшие площадки для размещения рекламы и время, в которое активнее всего ваша аудитория. Благодаря настройкам систему можно контролировать и настраивать ее по своим предпочтениям. Данное приложение ведет сразу несколько социальных сетей.

Все сервисы популярны и платные. **SMM-box** анализирует результаты работы, что является дополнительным плюсом. Сервис **Sociate** помимо автопостинга помогает с запуском и управлением таргетированной рекламы, в нем есть выгрузка отчетов со скриншотами и с помощью него можно запускать рекламу в Instagram. В плане большого выбора социальных сетей можно выделить **SMM-box**. Данный сервис обслуживает сразу 8 площадок, что удобно для администраторов, которые ведут группы сразу в 3–4 социальных сетях. В отличие от остальных, данный сервис предоставляет 14 дней бесплатного пользования, что очень удобно.

Сервисы, занимающиеся таргетингом:

1. **Церебро таргет**. В данном приложении находятся базы потенциальных клиентов. В нем можно найти своих конкурентов, аудиторию, для которой необходимо запустить рекламу, а также собрать список клиентов по номеру id. Приложение платное.
2. **GetTarget**. Этот сервис позволяет собрать данные пользователей, которые вступили в вашу группу и интересуются вашим сообществом. Также в программе можно посмотреть участников, которые вышли из сообщества. Таким образом, можно настроить и показывать рекламу только заинтересованным в этом людям, благодаря чему затраты существенно снижаются. Приложение платное, но есть бесплатный тестовый период в течение первых 14 дней использования.

Оба сервиса важны и стоит пользоваться обоими. В чем же отличие? **Церебро** помогает найти свою аудиторию, которой нужна ваша реклама. Вы сможете отследить подписчиков, их профили, найти именно активную аудиторию и обезопасить себя от ботов. Есть и огромное преимущество в возможности найти людей по дню рождения. У **GetTarget** своя функция и тоже немаловажная – он отслеживает пользователей, которые только что вступили в группу. Соответственно, можно настроить показ рекламы для людей, напрямую заинтересованных в вашем сообществе.

Подводя итог, важно отметить, что для продвижения группы «ВКонтакте» необходимо иметь качественный и уникальный контент. По этой причине сервисы-автопостинги не подойдут тем, кто хочет отличаться от конкурентов. Использование данных сервисов приводит к тому, что посты в нескольких группах повторяются, а это вызывает негативную реакцию у

подписчиков. Сервисы Церебро и GetTarget обязательно нужно использовать, если вы хотите привлечь к себе целевую аудиторию, которая заинтересована в вас и в том, что вы продвигаете. Сервисы-анализаторы лучше протестировать самим, посмотреть, какой удобен и подойдет по параметрам именно для вас. Но если вы ориентируетесь именно на отзывы и деньги вам не важны, то лучшим является сервис Jagajam.com. Лидеры мнений выбирают его [2, 3].

Литература

1. Румянцев Д. Продвижение бизнеса ВКонтакте. Новые практики и технологии. – Изд-во: Питер, 2016. – 400 с.
2. Public Relations в социальных сервисах сети Интернет [Электронный ресурс]. – Режим доступа: <http://bibliofond.ru/view.aspx?id=584133#1>, своб.
3. Халилов Д. Продвижение бизнеса ВКонтакте. – М.: Манн, Иванов и Фербер, 2013. – 376 с.



Фролова Мария Владимировна

Год рождения: 1994

Факультет информационных технологий и программирования,
кафедра информационных систем, группа № М4205

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: frolova.maria.vl@gmail.com

УДК 004.42

АНАЛИЗ МЕТОДОВ ОБМЕНА ДАННЫМИ ПРИ КЛИЕНТ-СЕРВЕРНОМ ВЗАИМОДЕЙСТВИИ

Фролова М.В.

Научный руководитель – к.т.н., доцент Хлопотов М.В.

В работе описаны технологии, обеспечивающие актуальность получаемой пользователями информации. Рассмотрены основные методы интенсивного обмена данными при взаимодействии клиента и сервера в веб-приложениях. Выявлены достоинства и недостатки каждого метода.

Ключевые слова: сервер, клиент, частый опрос, длинный опрос, WebSocket.

В настоящее время широко используются различные веб-приложения, которые позволяют онлайн бронировать места в гостиницах и кинотеатрах, заказывать авиа- и железнодорожные билеты, такси, совершать покупки по Интернету. Основным критерием при проектировании и разработке таких веб-приложений является актуальность информации. Она достигается с помощью частого обмена данными между клиентом (браузером, в котором визуально отображается информация) и сервером, который хранит и обрабатывает данные [1–3].

В современном мире информационных технологий для мгновенного и частого обмена данными существует множество способов. Наиболее известные из них:

- частый опрос (polling);
- длинный опрос (longpolling);
- WebSocket.

Каждая из этих технологий позволяет делать запросы к серверу, получать ответы и отображать пришедшие данные на клиентской стороне без перезагрузки страницы и с минимальным изменением HTML-структуры. Сравнительная характеристика методов представлена в таблице.

Таблица. Сравнительная характеристика методов обмена данными

	Высокая временная точность	Небольшой трафик	Двустороннее соединение	Поддержка во всех браузерах	Время жизни канала без ограничений
Частый опрос	–	–	–	+	–
Длинный опрос	+	+	–	+	–
WebSocket	+	+	+	–	+

Частый опрос – способ, основанный на постоянном обращении клиента к серверу за новой информацией. С заданной временной периодичностью с клиентской стороны отправляется запрос на сервер о наличии событий. Сервер формирует ответ на каждый запрос клиента независимо от наличия новых данных. Запрос выполняется с использованием подхода **AJAX**, который заключается в «фоновом» обмене данными браузера с сервером, что позволяет выполнять обмен данными без полной перезагрузки веб-страницы.

Основные достоинства частого опроса – простота реализации и поддержка во всех современных браузерах. Недостатки – задержки во времени между событием и уведомлением, лишний трафик и запросы на сервер.

Длинные опросы – отличная альтернатива частым опросам. Они также удобны в реализации, но при этом сообщения доставляются без задержек.

Клиент отправляет запрос на сервер. Сервер не отвечает, и соединение не закрывается, пока не произойдет событие, и новые данные не будут получены. Когда данные появляются, сервер отвечает на запрос, и соединение закрывается. Браузер делает новый запрос. Так продолжается до тех пор, пока пользователь не закроет веб-страницу, с которой отправляются запросы.

Длинные опросы хорошо подходят для тех случаев, когда сообщения от сервера приходят редко. По сравнению с частым опросом данная технология позволяет снизить нагрузку на сервер, уменьшить трафик и значительно повысить точность событий. Также как и частые опросы, длинные опросы работают во всех браузерах с поддержкой JavaScript.

WebSocket – новый протокол, работающий поверх TCP на одном уровне с HTTP и позволяющий создавать двунаправленную связь. Протокол определяется стандартом **RFC 6455** и позволяет создавать нешифрованное и зашифрованное соединения.

Клиент устанавливает TCP-соединение к серверу, формирует особый HTTP-запрос и отправляет его. Если соединение возможно, то после отправки сервером ответа WebSocket-соединение считается установленным, клиент и сервер начинают двунаправленный обмен сообщениями по этому же TCP-соединению. Больше HTTP не используется в протоколе, идет обмен данными на специальном протоколе WebSocket.

Формат пакета (фрейма) данных имеет компактную структуру благодаря тому, что служебная информация занимает всего два дополнительных байта (один в начале и в конце сообщения).

Протокол WebSocket не имеет ограничений на время жизни в неактивном состоянии, соединение может находиться в неактивном состоянии и не требовать ресурсов.

Данная технология дает возможность установить двустороннее соединение, снизить используемый трафик и поддерживать время жизни канала без ограничений. WebSocket необходим при разработке веб-приложений с интенсивным обменом

данными, комплексных приложений с множеством различных асинхронных блоков на странице, а также для создания кросс-доменных приложений: ограничения в протоколе вводятся не по принципу «из того же источника», а из «разрешенного источника», и определяются на сервере. Протокол WebSocket не поддерживается в устаревших браузерах, и для его реализации необходим WebSocket-сервер, что может вызвать некоторые трудности.

Таким образом, использование современных подходов для мгновенного и интенсивного обмена данными при клиент-серверном взаимодействии позволяет существенно повысить производительность интерактивных веб-приложений. Выбор метода зависит от поставленных задач и имеющихся возможностей.

Литература

1. COMET с XMLHttpRequest: длинные опросы [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/xhr-longpoll>, своб.
2. The WebSocket API [Электронный ресурс]. – Режим доступа: <https://www.w3.org/TR/websockets/>, своб.
3. The WebSocket Protocol [Электронный ресурс]. – Режим доступа: <https://tools.ietf.org/html/rfc6455>, своб.



Хрусталева Мария Максимовна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: lovelace@inbox.ru

УДК 30.308

ПОЛИТИЧЕСКАЯ ИНФОГРАФИКА

Хрусталева М.М.

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 рассмотрено понятие «политическая инфографика». Изучены и описаны ее взаимосвязи со смежными областями знаний. Обоснована необходимость исследований в данной области, а также показаны преимущества графического способа представления информации, как в политике, так и за ее пределами.

Ключевые слова: инфографика, политическая инфографика, визуализация, графическое представление информации, графический дизайн.

Инфографика – визуализация данных или идей, целью которой является донесение сложной информации до аудитории быстрым и понятным образом. Средства инфографики помимо изображений могут включать в себя графики, диаграммы, блок-схемы, таблицы, карты, списки [1].

В современном мире человек сталкивается с огромным количеством потребляемой информации.

Общеизвестным фактом является то, что чаще всего графическая информация более понятна и долговечна в памяти человека, нежели текстовая. Нам сложно запомнить несколько страниц текста, но если этот же текст мы визуализируем в картинку, то, скорее всего, запомним его без особых усилий.

Обычно люди хотят, как можно быстрее получить нужную им информацию, и использование визуалов (визуально оформленной информации – графиков, диаграмм, схем) является отличным способом удовлетворения этой потребности.

Инфографика – универсальное средство для распространения концептуальной информации.

Инфографику активно используют в совершенно разных областях, начиная от науки и статистики и заканчивая журналистикой и образованием. Отдельно стоит отметить ее применение в сфере политики.

Политические исследования тесно сопряжены с социологическими. Представление анализа – завершающий этап социологического исследования. Формы представления могут быть различными от текстового описания до графического представления. Современный способ визуального или наглядного представления информации – инфографика.

Существует несколько типов представления инфографики – статичная и динамическая. Статичной называют инфографику без анимационных элементов. Динамическая инфографика содержит анимацию и интерактивные элементы, такие как многоуровневая навигация, интегрированные фото- и видеоматериалы, звуковое сопровождение и т.п.

В сфере политики в подавляющем большинстве используется и создается статическая инфографика. Безусловно с ее помощью проводится и, более того, становится более эффективной политическая агитация.

При анализе методов, реально применяемых в средствах массовой информации для оказания политического влияния на население, очевидным становится, что представители политической журналистики неплохо знают психологию и весьма осмысленно применяют ее при подготовке агитационных и пропагандистских материалов. Этот факт вместе с визуальным представлением информации в виде инфографики способствует ее наилучшему усваиванию у потребителя.

Исходя из этого, политическая инфографика как продукт графического дизайна должна опираться на следующие принципы [2]:

- системность;
- лаконичность;
- структурированность и логичность организации информации, обусловленной функцией;
- образность;
- эстетическая привлекательность.

Надо сказать, что в нашей стране данный вид представления политической информации не очень широко распространен на сегодняшний день, в отличие от западных и, в частности, американских СМИ.

В ходе исследования было выявлено, что политической инфографике как понятию уделяется незаслуженно малое количество исследований, как в России, так и за рубежом. Между тем тема является очень актуальной. Целью дальнейших исследований является попытка восполнить белые пятна в данной предметной области.

Литература

1. Крам Р. Инфографика. Визуальное представление данных. СПб., 2015. – 37 с.
2. Яу Н. Искусство визуализации в бизнесе. Как представить сложную информацию простыми образами / Н. Яу. – СПб.: Манн, Иванов и Фербер, 2013. – 352 с.



Чебыкина Дарья Андреевна

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: chebyk@ro.ru

УДК 615.892

МЕТОДЫ АВТОМАТИЗАЦИИ ПОДГОТОВКИ ТЕХНИЧЕСКОГО ЗАДАНИЯ ПО ВЫПУСКУ ПОЛИГРАФИЧЕСКОЙ ПРОДУКЦИИ

Чебыкина Д.А.

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 решения в области автоматизации процессов допечатной подготовки полиграфической продукции. Предложена методика автоматизации подготовки требований к заказу клиентом путем оперативного взаимодействия с ним издательства. Рассмотрена реализация данной методики на примере создания онлайн-сервиса на сайте фирмы.

Ключевые слова: автоматизация, верстка, допечатная подготовка, полиграфия, реклама.

Несмотря на огромную конкуренцию на современном рынке рекламного бизнеса, рекламная полиграфия и полиграфические услуги пользуются популярностью у рекламодателей. На бумажных носителях размещают свою рекламу как крупные компании, так и небольшие фирмы.

Особое место в данной сфере бизнеса занимают периодические печатные издания, а именно, рекламные газеты и журналы. Допечатная подготовка данного вида периодики требует обработки огромного объема рекламной информации и для обеспечения оперативной работы издательства нуждается в современных средствах автоматизации.

Таким образом, задачей исследования являлся поиск и анализ путей оптимизации, ускорения и упрощения процесса верстки печатного издания, помогающих уменьшить затраты человеческого труда.

Процесс допечатной подготовки полиграфической продукции – это совокупность слаженных и взаимосвязанных между собой операций. Для того чтобы обозначить задачи автоматизации данного процесса, необходимо учесть факторы, обуславливающие результативность работ по подготовке издания на разных этапах процесса.

С.А. Назарова в своей работе исследовала факторы, которые влияют на процесс верстки издания и определяют ее результативность [1]. Среди них можно выделить факторы, в значительной степени поддающиеся контролю путем автоматизации процессов подготовки издания. Были выделены: объем материала, форма взаимодействия верстальщика с другими участниками процесса допечатной подготовки продукта, а также целая группа факторов, которые характеризуют оригинал-макет издания в качестве результата процесса верстки (различные сведения, уточняющие параметры верстки).

Что касается автоматизации верстки оригинал-макета и самих рекламных блоков, в области этой проблемы существует множество готовых решений, широко

представленных на рынке. В ряде работ исследователи занимаются вопросами усовершенствования основной программы путем подключения к ней различных плагинов [2, 3].

Другой фактор, влияющий на результативность допечатной подготовки издания – форма взаимодействия участников процесса. Для автоматизации бизнес-процессов внутри издательства разработаны полноценные редакционно-издательские системы. Данные системы предоставляют большие возможности для упрощения процесса верстки, однако они не решают задачу автоматизации процессов, происходящих между заказчиком (рекламодателем) и издательством. Среди таких процессов: согласование места размещения рекламы, обсуждение требований к верстке, утверждение готового результата, изменение рекламных данных и другое. Взаимодействие сотрудников компании с заказчиком происходит при личных встречах, телефонных переговорах или путем отправки сообщений на электронную почту. В случаях, когда требуется оперативное обновление информации, такой подход малоэффективен.

Вопросы эффективности коммуникации между клиентом и издательством рассматривали в своей работе Т.С. Попова и А.В. Бизюк [3]. В данной статье проблема рассмотрена на примере взаимодействия фирмы и заказчика по поводу изготовления печатной продукции (меню). Заказчик должен был иметь возможность еженедельно обновлять некоторую рекламную информацию, при этом издательство должно придерживаться старого оформления (дизайна) основных элементов. В связи с этим авторами статьи предложена технологическая схема автоматической верстки печатного издания с переменными данными. Так, сокращается общее время подготовки издания, так как она требует лишь обновления переменных данных макета, полученных от заказчика. Методика направлена на взаимодействие верстальщика и заказчика и сокращение времени на составление требований к изданию. Однако автор подчеркивает, что исследование применимо в области подготовки малых тиражей.

Таким образом, все рассмотренные методы направлены на автоматизацию процессов на этапе подготовки издания. Методы могут быть применимы в различных случаях, а в совокупности с использованием современных специализированных программных продуктов помогут оптимизировать работу издательства. Слабо исследованными остаются возможности издательства по автоматизации процессов взаимодействия с заказчиком, а также действий самого заказчика при составлении требований к продукту. В связи с этим была предложена собственная методика, направленная на совершенствование работы с клиентами.

Методика предполагала разработку онлайн-сервиса для взаимодействия рекламодателя и издательства в сфере оказания услуг по размещению рекламной информации на страницах рекламного издания. На сайт фирмы предлагалось размещать шаблон каждого нового выпуска продукта. В макете будут обозначены свободные рекламные блоки и требования к размещаемому графическому файлу. Таким образом, заказчик сам сможет выбрать любой свободный блок под размещение своей рекламы, предварительно увидеть его на странице издания (в случае, когда фирма имеет готовый рекламный блок) или забронировать место под рекламу, заказав ее верстку в самом издательстве, тогда файл будет прикреплен позже.

После совершения оплаты, верстальщик издания утвердит рекламу, а заказчик сможет оперативно вносить изменения вплоть до сдачи выпуска на печать. Так макет издания будет постепенно наполняться рекламными изображениями, а все участники процесса смогут за этим наблюдать.

При использовании данного подхода заказчик сможет напрямую участвовать в процессе верстки. Так верстальщику не придется учитывать его требования. Кроме этого, данное решение поможет автоматизировать и другие процессы: издательству не

придется заниматься разверсткой рекламы, так как рекламные блоки уже будут размещены на нужных местах.

Таким образом, в ходе данного исследования были рассмотрены готовые решения по автоматизации некоторых процессов допечатной подготовки полиграфической продукции. В результате было предложено собственное решение в виде разработки специализированного инструмента взаимодействия заказчика и издательства. Внедрение данного решения в деятельность фирмы может способствовать расширению клиентской базы и увеличению прибыли путем достижения объективных преимуществ перед фирмами-конкурентами: удобный инструмент рекламодателя для описания требований к заказу, минимальное время его обработки и оперативное взаимодействие с издательством [4].

Литература

1. Попова Т.С., Бизюк А.В. Методика автоматизации процесса верстки печатного издания с переменной графической информацией // Восточно-Европейский журнал передовых технологий. – 2011. – Т. 1. – № 10. – С. 24–25.
2. Манаков В.П., Пулавская О.В. Автоматизация процесса верстки печатного издания // Информационные технологии. – 2011. – Т. 6. – № 2. – С. 54.
3. 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.
4. Назарова С.А. Поддержка решений по выбору программно-технических средств верстки издания // Научные труды Винницкого национального технического университета. – 2015. – № 3. – С. 46–54.



Чердынцева Юлия Владимировна

Год рождения: 1995

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107с

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: julia.mk63@gmail.com



Сокуренко Юрий Андреевич

Год рождения: 1937

Факультет методов и техники управления «Академия ЛИМТУ»,

к.т.н., доцент

e-mail: kpd@limtu.ru

УДК 004

АНАЛИЗ ТЕХНОЛОГИЙ РАЗРАБОТКИ ВЕБ-АНИМАЦИИ

Чердынцева Ю.В., Сокуренко Ю.А.

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 проанализированы технологии разработки веб-анимации, рассмотрены достоинства и недостатки каждой из них. Приведены выводы о возможностях создания анимации на Flash и HTML5.

Ключевые слова: веб-анимация, HTML5, Flash, Canvas, WebGL.

Веб-анимация позволяет оживить дизайн сайта, сделать его более интересным для пользователя и служит хорошим дополнением к остальным элементам. Использовать ее для украшения сайтов начали очень давно – все началось с файлов формата .gif, которые не требовали специальных программ для просмотра и поддерживались почти всеми браузерами. Тогда это служило больше украшением внешнего вида, но сегодня при помощи JavaScript и CSS есть возможность улучшить удобство использования сайта, привлечь внимание пользователя к важным деталям.

Очень большую популярность получила Flash-анимация – технология, выпущенная в 1996 году компанией Macromedia и позволяющая объединить звук, текст, графику и анимацию. Эта платформа стала основой веб-анимации и долгое время была стандартом разработки интерактивных веб-сайтов. Активно использовались такие типа Flash-анимации, как Flash-баннеры, Flash-заставки, Flash-элементы, также были очень популярны Flash-сайты, но у них были существенные недостатки – долгая загрузка и сложность поисковой оптимизации.

Скорее именно коммерциализация интернета, т.е. необходимость извлечения прибыли, и как следствие, обеспечение конкурентоспособной привлекательности и необходимость рекламы своих товаров и услуг на этом новом рынке вызвала к жизни такие инструменты как Flash. До тех пор, пока интернет служил лишь источником дополнительной информации ученым, студентам, программистам и представителям технических профессий, возможностей HTML вполне хватало. И хотя появившиеся инструменты веб-дизайна и веб-программирования, такие как CSS, JavaScript, PHP внесли определенную динамичность и интерактив на web-сайты, этого оказалось недостаточно для развлекательного контента. В связи с этим появление на рынке такого инструмента, который позволял придавать красочность и динамичность сайту, а также возможность использовать множество звуков и звуковых эффектов и интересные графические анимации оказалось как нельзя кстати [1].

С появлением стандарта HTML5 популярность использования Flash-технологий значительно снизилась. Одним из главных преимуществ нового стандарта является его адаптивность и способность работать на мобильных устройствах, так как по статистике каждый пятый просмотр веб-страниц совершается с мобильного устройства. Помимо этого в 2014 и 2015 годах во Flash было обнаружено 457 уязвимостей, из-за чего около половины сайтов отказались от его использования, а браузеры начали блокировать. В то же время компания Google выпустила конвертер Flash-баннеров в HTML5, потому что многие телефоны не поддерживали воспроизведение Flash-контента.

Даже несмотря на то, что HTML5 был хорошо известен среди веб-разработчиков в течение нескольких лет, он стал основной темой СМИ только в апреле 2010 г. После этого глава компании Apple Inc. Стив Джобс написал публичное письмо, заголовок которого гласил: «Мысли по поводу Flash», где он заключил, что с разработкой HTML5 нет больше необходимости смотреть видеоролики или использовать другие виды приложений с помощью Adobe Flash. Но разработчики должны принимать во внимание различия браузеров и необходимость поддержки различных частей стандартов, равно как и функциональные различия между HTML5 и Flash [2].

Для сравнения можно выделить следующие преимущества HTML5 перед Flash-технологией:

- баннеры HTML5 адаптивны, не требуют подключения дополнительных плагинов для отображения и могут интегрироваться с приложениями, социальными сетями и т.п.;

- баннеры формата HTML5 используют меньше ресурсов, следовательно, не замедляют загрузку сайтов;
- наличие семантической разметки: HTML5-баннеры можно отслеживать с помощью Google Analytics;
- Flash-объявления могут быть заблокированы Firewall-ом или блокировщиками рекламы.

В настоящее время отношение к Flash-анимации и Flash-технологии в целом неоднозначно. С одной стороны – это одна из самых популярных и востребованных мультимедийных технологий. Ее популярность, обусловлена не только тем, что движущиеся объекты придают сайту эффектный вид и привлекают посетителя куда больше, чем статичные, но также и тем, что Flash-анимация в разы успешнее выполняет рекламные функции. Ведь информация, носящая динамический характер, больше обращает на себя внимание 160 посетителей сайта, чем просто рекламный текст или статичный баннер, лучше усваивается и дольше хранится в памяти – таковы особенности нашего восприятия [3].

Вполне возможно, что HTML5 постепенно вытеснит Flash из интернета, и Flash полностью переместится в другие отрасли, например, заняв нишу инструмента создания интерактивных презентаций [4].

Уже много лет между веб-разработчиками ведутся споры о том, что же лучше использовать – HTML5 или Flash, и окончательным завершением этой войны стал отказ компании Google от использования технологии Flash со 2 января 2017 года.

Одним из важных инструментов HTML5 является Canvas (холст) – инструмент для работы с двумерными графическими объектами при помощи кода JavaScript. Он используется для отображения анимации, графиков, диаграмм, с помощью него можно рисовать простейшие фигуры или создавать игры. В браузере он отображается как пустой холст для рисования, который создается при помощи тега `<canvas>`. У элемента `<canvas>` указаны три атрибута – `id`, `width` и `height`. Атрибут `id` используется для присваивания имени, которое требуется для идентификации кодом JavaScript, атрибуты `width` и `height` указывают ширину и высоту холста в пикселах.

Отдельно стоит упомянуть библиотеку WebGL (Web-based Graphics Library), которая предназначена для визуализации интерактивной 3D-графики и поддерживается настольными персональными компьютерами, мобильными браузерами и платформами. Она использует Canvas, элемент HTML5, для интеграции в web-страницу и позволяет создавать реалистичные 3D-игры, эффекты и сцены без помощи дополнительных плагинов.

Таким образом, веб-анимация прочно вошла в историю разработки сайтов и продолжает совершенствоваться, делая работу пользователя все более удобной и дополняя визуальную составляющую сайтов [5].

Литература

1. Филимоненкова Т.Н., Широков А.С. Flash-технология и web-дизайн // Сб. трудов конференции «Информационные системы и технологии в моделировании и управлении». – 2016. – С. 272–276.
2. Новожилов А.В., Акулов Г.С. Поддержка браузерами технологий HTML5 и CSS3 // Вестник Тамбовского университета. Серия: естественные и технические науки. – 2014. – Т. 19. – № 2. – С. 663–665.
3. Перегудова М.М. Анимация Flash в современном веб-дизайне // Сб. трудов конференции «Проблемы управления в социально-экономических и технических системах». – 2016. – С. 159–163.

4. Зятькова Л.К., Комиссарова Е.В., Колесников А.А. Современные web-технологии для создания интерактивных мультимедийных картографических произведений // Изв. вузов. Геодезия и аэрофотосъемка. – 2012. – № 2.1. – С. 95–98.
5. Смирнов А.И. MacromediaFlash: стратегии и использования на web-сайтах // Вестник Тамбовского университета. Серия: естественные и технические науки. – 2002. – Т. 7. – № 1. – С. 84–85.



Черепанов Дмитрий Михайлович

Год рождения: 1989

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: d-work89@mail.ru

УДК 004.92

**ТЕХНОЛОГИИ ТРЕХМЕРНОГО МОДЕЛИРОВАНИЯ ПРЕДМЕТОВ ИНТЕРЬЕРА
И МЕБЕЛИ**

Черепанов Д.М.

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 технологии трехмерного моделирования предметов интерьера и мебели. Описаны виды технологий трехмерного моделирования, их преимущества и недостатки.

Ключевые слова: технология, модель, трехмерное моделирование, мебель, интерьер.

В современном мире, где главенствующую роль во всех сферах жизни человека играет информация, все больше и больше возникает потребность в ее обработке для более полного восприятия. Одним из способов такой обработки стало моделирование.

Моделирование – построение и изучение моделей реально существующих объектов, процессов или явлений с целью получения объяснений этих явлений, а также для предсказания явлений, интересующих исследователя.

Моделировать можно:

- объекты (сооружения, игрушки);
- явления (грозовой разряд, землетрясение);
- процессы (развитие вселенной);
- поведение (при выполнении человеком действия ему предшествует возникновение в сознании модели будущего поведения).

Модель может содержать описание характеристик и свойств объекта.

В силу того, что понятие «модель» имеет большое число значений, в науке и технике не существует единой классификации видов моделирования: классификацию можно проводить по характеру моделей, по характеру моделируемых объектов, по сферам приложения моделирования (в технике, физических науках, кибернетике и т.д.).

Можно выделить следующие виды технологий трехмерного моделирования, которые могут применяться в том числе и для моделирования предметов интерьера и мебели:

- информационное моделирование;

- компьютерное моделирование;
- физическое моделирование;
- графическое и геометрическое моделирование.

Под информационным моделированием в данном случае следует понимать технологию, с помощью которой создается информационная модель объекта или явления в виде документов на носителях информации – бумаге, дисковых накопителях и файлов компьютерной системы в различных форматах.

Под компьютерным моделированием понимается весь набор способов, с помощью которых можно создать модель объекта, используя компьютерную систему.

В процессе физического моделирования можно создать материальную, осязаемую на ощупь модель объекта или явления. К таким способам моделирования относят, например, выточку модели ножки стола из деревянной заготовки на специализированном станке.

Графическое моделирование используется для наглядного отображения реального объекта с помощью его модели. Для этих целей используются рисунки, чертежи, различные изображения на экране компьютера, полученные, к примеру, в ходе компьютерного моделирования.

Все вышеперечисленные технологии являются различными самостоятельными видами представления моделей. Тем не менее, ни про одну из них нельзя сказать, что она не перекликается и никак не взаимодействует с другими технологиями моделирования. Например, для создания трехмерной модели стула на компьютере вам потребуются знания о его свойствах, таких, как длина, ширина, высота, цвет, материал, форма и другие. Список этих свойств может быть очень длинным. Для упрощения работы можно составить его на бумажном носителе в виде таблиц и рисунков (чертежей), ориентируясь по которым вам будет гораздо проще выполнить построение трехмерной модели на компьютере.

С физическим моделированием ситуация аналогичная. Можно выточить на станке из заготовки деталь вручную, смотря на образец. Но это будет долго и весьма не точно. А можно создать программу с выточкой детали по образцу компьютерной модели. И тогда станок все выточит сам. И, что самое главное, при желании – в любом количестве. Экономия времени и сил при этом будет велика.

Как видно из сказанного выше, моделирование – один из важных процессов в жизни современного общества. При этом от выбора технологии напрямую зависит количество затраченного времени и сил на построение модели. Следует отметить, что выбор технологии и способа создания модели определяется конкретной ситуацией и целями, которые зависят от потребностей и желаний людей.

Для целей работы на данный момент можно выделить следующие технологии создания трехмерных моделей:

- создание компьютерной модели в пакетах графического моделирования;
- создание компьютерной модели с помощью лазера и оптических средств;
- создание физической модели с помощью 3D-принтера;
- создание физической модели с помощью промышленных средств и технических инструментов.

Первый способ создания трехмерной модели использует специализированные пакеты графического моделирования и компьютерную систему [1, 2]. Итогом проектирования в данном случае станет как визуальная составляющая модели в виде отображаемого на экране компьютера изображения, так и электронный документ с описанием модели в виде компьютерного файла на носителе информации.

Преимуществом такого способа моделирования является очень большой простор для изменений как самой модели, так и ее частей. Причем изменению можно подвергнуть как технические свойства модели, так и отображение ее на экране. Таким способом можно смоделировать что угодно, начиная от обычного деревянного стула и до мягкого удобного кресла с кожаной обивкой. Все ограничено лишь возможностями компьютерной системы и навыками человека, создающего модель.

При этом следует отметить, что этот способ моделирования не лишен недостатков. Хотя он и позволяет моделировать практически любой объект, но с его помощью нельзя смоделировать то, на что не хватит ресурсов компьютерной системы. Конечно же, можно увеличить потенциал системы, расширив ее составляющие и докупив более мощные по производительности аппаратные части. Но это требует дополнительных финансовых затрат, равно как и дополнительного труда человека.

Таким образом, несмотря на то, что сама модель выходит бесплатной, сам процесс моделирования может выйти чрезвычайно дорогостоящим в зависимости от требований к модели.

Второй способ создания моделей использует систему оптических и лазерных устройств для получения информации об объекте для построения его модели на компьютере [3, 4]. Его главное преимущество в отсутствии прямого контакта с объектом моделирования. Это позволяет создавать модели объектов, находящихся в труднодоступных местах или хрупких объектов, которые могут подвергнуться разрушению и уничтожению в случае прямого, контактного анализа.

Такой способ очень важен, например, для предметов, имеющих важное историческое значение – старых книг, рукописей, картин, свитков и прочего, что под воздействием времени стало очень хрупким. Применительно же к теме статьи, этим способом можно смоделировать, к примеру, лепнину залов, старые люстры и канделябры, и многие другие детали интерьеров прошлого для целей реставрации в музеях и театрах.

Недостатки данного способа проявляются в относительной сложности расчета и проектирования модели, а также в погрешностях моделирования – модель может иметь много лишних деталей или же сильно нагружать компьютерную систему из-за выбранной точности моделирования.

Третий способ создания моделей использует для этих целей трехмерный принтер [5]. Этот принтер позволяет создавать объемную, осязаемую, трехмерную физическую модель объекта из различных материалов, будь то пластик, жидкое тесто или даже бетон. Трехмерные принтеры имеют различные размеры, от которых зависят характеристики создаваемой модели: минимальные и максимальные длина, ширина и высота готовой модели, что определяет точность моделирования.

От используемого материала и точности моделирования зависит цена трехмерного принтера, что можно отнести как к плюсу, так и к минусу, в зависимости от ситуации. В ходе создания модели таким способом стоит также учесть и стоимость трехмерной компьютерной модели, которая будет использоваться для печати на принтере.

Как видно из используемых в данной технологии материалов, таким способом можно создавать довольно широкий спектр моделей, как пластиковые фигурки и торты, так и детали технических устройств и даже целые здания. Однако нельзя создать абсолютно все. Некоторые предметы и детали интерьера таким способом создать не удастся. К примеру, та же кожаная обивка для кресла. Это связано с особенностями создания модели в рамках данной технологии. Следовательно, пока не вышла соответствующая технология, это также будет являться минусом данной технологии.

Четвертый способ создания моделей является, пожалуй, самым традиционным из всех. Он использует различные технические средства для создания трехмерной модели. Это может быть, к примеру, деревообрабатывающий станок, для создания модели объекта из заготовок. Или же набор инструментов для резьбы по дереву, металлу, и другим материалам.

Таким способом уже довольно много лет изготавливают составляющие для кухонного гарнитура, шкафов, комодов, столов и прочего. В последствии из этих составляющих, как из деталей конструктора, собирают мебель. В данном способе создание модели может происходить как автоматически, с использованием программируемых станков на заводе, так и вручную, используя труд рабочего, вытачивающего каждую деталь отдельно. От выбора метода работы зависит стоимость как всего процесса моделирования в целом, так и стоимость каждой конкретной изготовленной модели. Это можно отнести как к плюсу, так и к минусу данной технологии.

Как понятно из вышесказанного, все эти технологии потенциально можно использовать для создания трехмерных моделей предметов интерьера и мебели. И нельзя сказать, что какой-то из способов наиболее эффективен. Все зависит от конкретной ситуации и требований, предъявляемых к модели. Таким образом, можно сделать вывод о том, что нет особой разницы какую технологию использовать. Все это остается на суд и выбор человеку.

Литература

1. Кувшинов Н.С. Использование компьютерных технологий на основе графического пакета AutoCAD при конструировании изделий // Вестник ЮУрГУ. Серия: строительство и архитектура. – 2015. – Т. 15. – № 2. – С. 52–58.
2. Телегин И.В., Степанов А.С., Титов Н.А. AutodeskInventor. Чтение и детализирование сборочного чертежа // Научный Альманах. – 2015. – № 7(9). – С. 495–499.
3. Оленьков В.Д., Колмогорова А.О. Применение технологий трехмерного моделирования при воссоздании церкви Георгия Победоносца в селе Варламово Челябинской области // Наука ЮурГУ. – 2016. – С. 141–146.
4. Шеина С.Г., Упенников Д.К. Современные технологии, приборы и оборудование для трехмерного моделирования памятников архитектурного наследия // Интернет-журнал Науковедение. – 2015. – Т. 7. – № 3(28). – С. 134.
5. Батуева Э.Б., Мункуев В.Б. Проектирование объекта для 3D-печати // Культура и искусство. Поиски и открытия. – 2016. – С. 248–256.



Чуканбаева Надия Гарифжановна

Год рождения: 1986

Академия методов и техники управления «ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: chukanbaeva@gmail.com



Перепелица Филипп Александрович

Год рождения: 1977

Академия методов и техники управления «ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, ст. преподаватель
e-mail: autodesk@gmail.com

УДК 519.876.5

**ПРОБЛЕМАТИКА СОХРАНЕНИЯ ЦЕЛОСТНОСТИ ДАННЫХ
ИНФОРМАЦИОННОЙ МОДЕЛИ ПРИ ЕЕ ЭКСПОРТИРОВАНИИ ИЗ СРЕДЫ
РАЗРАБОТКИ В СРЕДУ ПРОСМОТРА**

Чуканбаева Н.Г., Перепелица Ф.А.

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

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе рассмотрены вопросы применения BIM-технологий, описана концепция информационной модели объекта, проанализированы проблемы использования информационной модели на этапе эксплуатации здания и предложены пути их решения.

Ключевые слова: BIM, проектирование, информационная модель, целостность данных.

Передовые технологии и совершенно новые подходы, к уже привычным и окончательно сформированным процессам, – это одна из самых характерных тенденций современности. Не стала исключением и строительная отрасль. Концепция информационного моделирования зданий – BIM (Building Information Modeling) – уже достаточно давно и успешно применяется специалистами по всему миру. Зародившийся в проектной среде и получивший широкое распространение при разработке новых объектов, этот подход ожидаемо быстро вышел далеко за установленные для него рамки, и сейчас информационное моделирование зданий – это намного больше, чем просто новый взгляд на проектирование. Скорее это принципиально иной подход ко всем этапам жизненного цикла объекта, а также к управлению его жизненным циклом в принципе.

Основа концепции BIM-технологий – это процессы, способы совместной работы всех специалистов, участвующих в жизненном цикле объекта, с информацией об объекте строительства. Причем эта информация не должна восприниматься как 3D-модель для визуализации проекта и создания проектной документации. Это ни в коем случае не аналог классического макета, лишь повторяющего геометрию объекта, просто выполненный на компьютере. На самом деле это по сути полноценный виртуальный дубликат здания со всеми его составляющими, с геометрическими и технологическими характеристиками всех конструкций, материалов и оборудования, входящих в его структуру, со всеми техническими паспортами, данными о поставщиках, и многим другим. При этом все эти данные об объекте не просто статично собраны воедино в некое подобие таблицы или справочника. Напротив, эта информация структурирована и параметризована таким образом, что она одновременно является характеристическим описанием модели, и корректировка этих данных с учетом существующих между ними зависимостей неизменно влечет за собой автоматическое изменение всей модели.

Информационная модель наполняется данными на протяжении всего жизненного цикла объекта, от проектирования до утилизации. Учитывая, что BIM-технологии

подразумевают концепцию коллективной работы, очевидно, что модель можно рассматривать как пространство, объединяющее в себе как давно существующие данные, так и новые знания, появляющиеся в процессе работы над проектом. Кроме того, этот подход предоставляет возможность организации полноценного обмена данными как между отдельными структурами и подразделениями компании, как и между компанией и внешними участниками разработки, например, поставщиками, подрядчиками и т.д. Таким образом, значительно упрощаются и синхронизируются такие задачи и процессы, как закупки, календарное планирование, документооборот и многие другие [1].

Во многих странах мира, как на Западе, так и на Востоке, уже порядка половины всех строительных проектов реализуются с применением концепции информационного моделирования. А в Великобритании, вообще, с 2015 г. законодательно принято, что все новое строительство должно выполняться только с применением BIM-комплексов [2]. В нашей стране, к сожалению, внедрение BIM-технологий пока находится на самом раннем своем этапе. Этому много причин, и исторических, и экономических, и даже психологических. Однако при всем при этом нельзя отрицать тот факт, что постепенно ситуация с использованием информационного моделирования в отечественных компаниях меняется к лучшему, наблюдается рост понимания топ-менеджерами того факта, что освоение BIM необходимо для дальнейшего развития бизнеса, и наиболее инновационные российские предприятия активно переходят на BIM и уже почувствовали преимущества от использования технологии. И если поначалу осознание необходимости внедрения информационного подхода к проектированию приходило к рядовым инженерам-проектировщикам, осваивавшим в свободное время передовые технологии и пытавшимся донести важность их применения до своего руководства, то сегодня ситуация поменялась, и процесс внедрения инициируется сверху, причем на самых высоких уровнях [3].

В частности, большим шагом к внедрению технологий информационного моделирования в области отечественного промышленного и гражданского строительства стал приказ Министра строительства и ЖКХ Российской Федерации № 926/пр от 29.12.2014 года «Об утверждении плана поэтапного внедрения технологий информационного моделирования в области промышленного и гражданского строительства», в котором был подробно описан «План поэтапного внедрения информационного моделирования в области промышленного и гражданского строительства». В правительстве ожидают, что в 2018–2019 годах государственный заказ в строительстве перейдет на технологию BIM [4]. Неудивительно, что после принятия упомянутого выше Плана резко возрос интерес участников российского строительного рынка к использованию технологии BIM, а значит, вопрос внедрения концепции информационного моделирования актуален как никогда. Учитывая относительную новизну этой технологии для нашей страны, логично прогнозировать появление большого количества новых стандартов, методик и технологий работы, причем как государственного, так и отраслевого уровня. И, к сожалению, не обойдется и без проблем внедрения и адаптации технологий к отечественным реалиям.

Опустим проблемы внедрения BIM, касающиеся этапов проектирования и строительства, и сконцентрируемся на этапе эксплуатации. Этот этап жизненного цикла здания представляется на данный момент наиболее важным для рассмотрения хотя бы только потому, что на момент начала активного внедрения информационного моделирования в России уже существует огромный жилой фонд, ранее почти полностью принадлежавший государству [3]. Сейчас он, находясь уже в разных формах собственности, требует постоянного обслуживания и капитального ремонта, однако за последние двадцать лет, в силу определенных обстоятельств, капитальным ремонтом

жилых зданий почти не занимались, к тому же в большинстве случаев техническая документация по зданиям частично или полностью отсутствует. В такой ситуации вполне логично было бы заменить бумажные паспорта зданий на их информационные модели. Конечно, информатизация сферы ЖКХ пугает очень многих, как минимум необходимостью создания моделей зданий. Тем не менее, если задуматься, то модель создается один раз и затем лишь изменяется и дополняется в соответствии с событиями, происходящими в процессе эксплуатации здания. Кроме того, в нашей стране достаточно большой объем от общего количества единиц жилого фонда представляет собой образцы типовой застройки. А значит, почти вполнину снизится количество уникальных моделей, как минимум в плане геометрии.

Однако есть другая проблема – предоставление сотрудникам ЖКХ технической возможности работы с информационной моделью здания. Ведь если представить себе усредненный «офис» среднестатистического жилуправления, то нетрудно понять, что вряд ли там будут стоять компьютеры, имеющие необходимую производительность и технические характеристики для просмотра полноценной информационной модели здания. Но суть даже не в этом. Официальное программное обеспечение для работы с информационными моделями стоит денег, которые даже если и выделяют, то достаточно долго будут доставлять до конечного получателя. И вот тут остро встает очевидный вопрос – как быть?

Выходом в данной ситуации видится использование программ-просмотрщиков, которые позволяют разворачивать на локальной машине модель здания, не предъявляя при этом слишком высоких требований к аппаратному обеспечению. Но даже такое упрощение не решает всех проблем, так как при этом остается открытым вопрос оплаты лицензий на программное обеспечение. Эта проблема может быть решена использованием бесплатных программ, например, продуктом Navisworks Freedom от компании Autodesk.

К сожалению, этот казался бы идеальный вариант решения проблемы имеет один маленький недостаток. Как показала практика, при экспорте модели из среды разработки для последующего развертывания на локальной машине средствами Navisworks Freedom происходит некоторое нарушение целостности данных, в частности, например, происходит разрыв связей между отдельным элементом геометрической модели и ссылками на документы, описывающие его.

Как вариант решения этого вопроса можно рассматривать разработку программных надстроек к среде Navisworks Freedom, позволяющих исправить текущую ситуацию. При таком подходе, однако, необходимо выяснить степень открытости среды к сторонней разработке и возможности самостоятельного создания надстроек. Это тема для размышлений, дискуссий и обсуждения с представителями компании Autodesk.

Вторым вариантом действий является написание некой внешней программы, которая выступала бы в качестве посредника между исходной средой разработки информационной модели, например, Autodesk Revit, и средой просмотра. Эта программа-посредник принимала бы данные о связях элементов модели с информацией из файлов других форматов, переписывала бы эти данные в один файл и передавала их в среду Navisworks в виде, который был бы удобен просмотрщику для максимально точного воспроизведения всех ссылок. А может, и не передавала бы их в Navisworks вообще, а хранила бы их в себе и активировалась как внешний диспетчер ссылок каждый раз, когда пользователю просмотрщика требовалось бы получить доступ к файлам описания геометрического объекта модели.

Каждый из предложенных вариантов имеет свои плюсы и минусы, но что важнее – каждый из них по-своему зависит от политики разработчика, в данном случае компании Autodesk. И окончательное решение по выбору того или иного метода будет

приниматься в большей степени на основе общения с представителями технической группы Autodesk и их ответов.

Тем не менее, одно ясно совершенно точно – в этот важный переходный момент, когда вопрос внедрения технологий информационного моделирования стоит уже на государственном уровне, сложно недооценить актуальность скорейшей адаптации всех участников жизненного цикла объектов строительства к работе с высокими технологиями. В частности, упрощение доступа работникам жилищно-коммунальных служб к цифровым данным о здании и всех его системах позволит ускорить процесс внедрения BIM-технологий в эту сферу, что, несомненно, послужит ощутимым толчком к расширению использования этих технологий.

Литература

1. Талапов В.В. Основы BIM: Введение в информационное моделирование зданий. – М.: ДМК Пресс, 2011. – 392 с.
2. BIM. Лучшая практика внедрения IT-технологий в градостроительной сфере [Электронный ресурс]. – Режим доступа: <http://ardexpert.ru/article/4239>, своб.
3. Козлов Н.А., Попова К.А. Проблемы внедрения технологий BIM проектирования в России [Электронный ресурс]. – Режим доступа: trts.esrae.ru/28-145, своб.
4. Гореткина Е. PLM-рынок в России: особенности, факторы влияния, перспективы // PC Week/RE. – 2013. – № 32-34. – С. 16–17.



Шадрин Дмитрий Михайлович

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: dm.shadrin@mail.ru



Погорелов Виктор Иванович

Год рождения: 1940

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, д.т.н., профессор

e-mail: kpd@limtu.ru

УДК 004.55

АНАЛИЗ ВОЗМОЖНОСТЕЙ СЕМАНТИЧЕСКОЙ ВЕРСТКИ ВЕБ-САЙТОВ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ

Шадрин Д.М.

Научный руководитель – д.т.н., профессор Погорелов В.И.

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе рассмотрены особенности семантической верстки веб-страниц, способствующие повышению доступности веб-сайта для лиц с ограниченными возможностями.

Ключевые слова: веб-сайт, доступность, семантическая верстка, ограниченные возможности, HTML, ARIA.

В 2014 году консорциумом W3C была утверждена новая спецификация языка разметки HTML5, одним из новшеств которого являлось возможность семантической верстки. Согласно одному из определений, семантика – изучение способа использования слов и передаваемых ими значений. Применительно к верстке веб-сайтов, под семантикой понимается наличие у HTML-тегов определенного смысла [1–4].

Ранее структура страницы определялась в основном требуемым дизайном. К примеру, тег `<i>` использовался только для создания курсивного текста, часто используемого для обозначения важных частей текста. В стандарте HTML5 предлагается иной подход к формированию структуры документа на основе смысла текста. Используя показанный ранее пример, для усиления значения текста используется тег `` (от англ. *emphasis*, акцентирование). Визуально `` ничем не отличается от `<i>`, но имеет определенный смысл и предназначение, которого лишен `<i>`.

В отличие от других нововведений HTML5, семантическая верстка не дает видимого результата для обычных пользователей. Большую выгоду от использования семантической верстки получают пользователи с ограниченными возможностями, которые используют альтернативное невизуальное представление веб-сайтов через программы-ассистенты, такие как программы для чтения с экрана. Благодаря наличию смысловых тегов, программы-ассистенты могут не просто произносить текст, а объяснять смысл тех или иных элементов и помогать пользователям в навигации по сайту.

Использование семантических тегов секционирования, таких как `<header>`, `<main>`, `<article>`, `<section>`, `<footer>` и других, является самым базовым способом адаптации для программ-ассистентов. Это позволяет программам-ассистентам осуществлять быструю навигацию между контентом. К примеру, ассистент может по запросу пользователя перейти к следующей статье, если текущая оказалась не интересной для пользователя. Это позволяет использовать универсальные методы навигации независимо от просматриваемого сайта.

Для улучшения восприятия основного контента страницы, следует использовать теги оформления контента, такие как цитаты `<blockquote>`, разъяснения аббревиатур `<abbr>`, списки определений `<dl>` и другие. Наличие такой разметки позволяет точнее донести содержимое страницы в альтернативных представлениях, например, изменять интонацию или голос диктора при озвучивании цитат.

Наиболее сложными для адаптации для лиц с ограниченными возможностями является интерактивные веб-приложения, содержащие интерактивные элементы, формы и медиа-контент. Для решения проблемы доступности таких веб-страниц внутри консорциума W3C была создана отдельная рабочая группа, которая занимается разработкой стандарта WAI-ARIA (Web Accessibility Initiative – Accessible Rich Internet Applications). Этот стандарт направлен на предоставление возможности полноценного использования веб-страниц людьми с ограниченными возможностями. Использование стандарта WAI-ARIA приносит наибольшую выгоду на веб-страницах с меняющимся содержимым, а также на страницах с нестандартными элементами управления.

Основной задачей стандарта ARIA является предоставление программам-ассистентам дополнительных сведений об элементах страницы. Стандарт допускает использование различных языков для верстки интерфейсов, однако применительно к

веб-разработке, используется только совместно с HTML. Стандарт предлагает два типа информации для программ ассистентов: роли и атрибуты. Роли сообщают ассистенту предназначение элемента, атрибуты сообщают некоторые свойства элемента.

Роль элемента несет информацию о функциональном предназначении элемента. Для добавления к элементу информации о его роли используется атрибут `role`. Большинству стандартных компонентов, таких как кнопка или ссылка, не требуется указывать роль явно. Применительно к блокам контента, таким как шапка или подвал, новые семантические теги HTML5 уже имеют предопределенные роли. К примеру, обе следующие строки приведут к одинаковому результату.

```
<footer> Нижний колонтитул </footer>  
<div role="contentinfo">Нижний колонтитул</div>
```

Однако предпочтительным является использовать семантические элементы HTML5. Благодаря ролям программа-ассистент позволяет пользователю быстро переходить к различным регионам страницы, например, быстро перейти к навигации сайта или, наоборот, пропустить меню и перейти к основному контенту.

Помимо навигационных ролей, существуют роли, описывающие поведение нестандартных веб-компонентов. К примеру, часто используется представление контента в виде вкладок. Язык разметки HTML не имеет встроенных механизмов реализации такого поведения. Стандарт ARIA предлагает роли `tab` для управляющих элементов и `tabpanel` для самого содержимого вкладки, что позволяет программе-ассистенту объяснить пользователю механизм работы вкладок и переключать их.

```
<ul role="tablist">  
<li>  
<a role="tab" aria-controls="tab1" aria-selected="true" href="#tab1" >  
Вкладка 1  
</a>  
</li>  
<li>  
<a role="tab" aria-controls="tab2" aria-selected="false" href="#tab2" >  
Вкладка 2  
</a>  
</li>  
</ul>  
<div id="tab1" role="tabpanel">Содержимое вкладки 1</div>  
<div id="tab2" role="tabpanel">Содержимое вкладки 2</div>
```

Атрибуты элементов передают ассистентам дополнительную информацию о предназначении элементов и возможных действиях над ними. Атрибуты ARIA добавляются в разметку таким же образом, как и роли. Существует множество различных атрибутов ARIA. Все они используют префикс `aria-` и условно делятся на две группы: свойства и состояния. Значения атрибутов-свойств, как правило, остаются неизменными, а значения атрибутов-состояний изменяются в результате взаимодействия с пользователем.

При просмотре пользователем с ограниченными возможностями веб-страниц, программа ассистент озвучивает имена элементов, чтобы пользователь понимал, что в данный момент просматривает или какое поле формы заполняет. В чистом HTML для задания имен используются атрибуты `title`. Стандарт ARIA расширяет семантические возможности именования элементов. Каждый элемент может иметь доступное имя (`accessible name`) и доступное описание (`accessible Description`). Имя предполагает короткое название, а описание более подробное. Для определения доступного имени может применяться атрибут `aria-label`. Значение атрибута не отображается в обычных браузерах. Если же имя элемента должно быть отображено на экране обычного

браузера, тогда должен применяться атрибут `aria-labelledby` с указанием идентификатора элемента, который описывает данный элемент. Он позволяет программам-ассистентам соотнести элемент и его название, содержащееся в другом элементе. Для задания доступного описания существует аналогичный атрибут `aria-describedby`. Следующий пример иллюстрирует текстовое поле ввода, для которого определены доступное имя, не отображаемое на экране, и доступное описание, видимое обычным пользователям.

```
<labelid="address">Введите адрес, куда следует доставить заказ</label>  
<input type="text" aria-label="address" aria-describedby="address">
```

В случае, если доступное имя и (или) доступное описание не указаны, программа-ассистент пытается подобрать подходящее значение из других имеющихся атрибутов, к примеру, `title` или `placeholder`.

Стандарт ARIA предлагает возможности для повышения удобства заполнения форм. Атрибут-состояние `aria-invalid` показывает, что пользователь ввел некорректные данные или допустил ошибку. Атрибут-состояние `aria-disabled` сообщает о неактивности элемента формы. Атрибут `aria-readonly` показывает, что поле недоступно для редактирования. Несмотря на то, что многие из этих элементов имеют прямые функциональные аналоги в HTML, оба типа атрибутов можно применять совместно.

Все вышеперечисленные возможности семантической верстки уже работают во всех современных браузерах и поддерживаются программами-ассистентами. Наличие семантики в элементах позволяет пользователям с ограниченными возможностями полноценно использовать веб-страницы наравне с обычными пользователями.

Литература

1. HTML 5.1: 3. Semantics, structure, and APIs of HTML documents [Электронный ресурс]. – Режим доступа: <https://www.w3.org/TR/html51/dom.html#elements-semantic>, своб.
2. Accessible Rich Internet Applications (WAI-ARIA) 1.0 [Электронный ресурс]. – Режим доступа: <https://www.w3.org/TR/wai-aria/>, своб.
3. ARIA in HTML [Электронный ресурс]. – Режим доступа: <https://www.w3.org/TR/html-aria/>, своб.
4. WAI-ARIA Authoring Practices 1.1 [Электронный ресурс]. – Режим доступа: <https://www.w3.org/TR/wai-aria-practices>, своб.



Шакирзянова Евгения Ильдаровна

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4105

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: jenya.1994@inbox.ru

УДК 004.588

ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ КАК ИНСТРУМЕНТ ПОВЫШЕНИЯ КАЧЕСТВА ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЕТЕЙ ДОШКОЛЬНОГО И НАЧАЛЬНОГО ШКОЛЬНОГО ВОЗРАСТА

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 исследована актуальность распространения технологий дополненной реальности, в частности, в образовательной сфере, проанализированы существующие зарубежные и отечественные решения, изучены библиотеки, платформы и AR-фреймворки, описаны этапы создания дополненной реальности для образовательных целей.

Ключевые слова: дополненная реальность, образование, приложение, виртуальная реальность.

XXI век – эра информационных технологий, время крупных цифровых открытий, глобальных технических прорывов в совершенствовании производственных процессов и систем.

В эпоху тотальной информатизации вопрос культуры общения людей с техническими устройствами актуален как никогда. На сегодняшний день разработано множество методик и технологий [1, С. 129], направленных на адаптацию пользователей к интерфейсам информационных систем. Одна из них – технологии дополненной реальности, иначе AR-технологии (Augmented Reality), позволяющие существенно расширить область данных, воспринимаемых человеком.

Термин «дополненная реальность» был предложен исследователем Томом Коделом в 1990 году [2]. В отличие от виртуальной реальности (VR-технологии), представляющей собой искусственную среду, созданную техническими средствами, в которой человек ощущает себя близко к тому, как он себя ощущает в реальном мире [3], дополненная реальность использует существующее пространство вокруг пользователя и накладывает поверх него новую информацию. Процесс переноса цифровой информации в реальный мир осуществляется при помощи устройства, оснащенного камерой. Для просмотра и воспроизведения видеосигнала, представленного виртуальными объектами, необходима специальная программа, считывающая закодированную картинку. Зачастую элементы AR-технологий – это видео- и аудиоматериалы, 3D-модели, а также текстовый контент.

Широкое распространение и актуальность технологий дополненной реальности, в частности, в образовании, связано с тем, что с каждым днем растет процент детей, пользующихся смарт-устройствами в основном для игровых и развлекательных целей. Борьба с возникшей проблемой бессмысленна. Целесообразное решение – сделать время, проводимое детьми с телефоном или планшетом, максимально полезным. Применение AR-технологий в образовательных организациях дошкольного образования и в начальной школе позволит в первую очередь повысить не только наглядность предлагаемого материала для изучения, но и качество восприятия информации через интерактивное взаимодействие детей с ним.

Процесс создания дополненной реальности включает несколько этапов:

1. построение виртуальной модели изображения;
2. связь виртуальной информации и статичного изображения при помощи точек;
3. отслеживание точек и построение траектории их движения;
4. написание приложения для распознавания объектов;
5. тестирование и проверка работоспособности системы.

Первый этап – создание виртуальной модели реального изображения. Для связи двух объектов выделяют специальные точки на самом изображении, перемещение которых будет отслеживаться. Из-за различного масштаба точек используют пирамиду изображений, где каждая последующая картинка уменьшается в два раза. При отслеживании точек строится траектория их движения, которая позволяет определить положение камеры в пространстве. Автоматизация и стабилизация работы достигается путем добавления новых точек слежения.

Написание приложения – второй этап работы по созданию дополненной реальности. В данном процессе необходимо использование специальных AR-библиотек с открытым API. Самыми популярными AR-фреймворками среди мобильных разработчиков на сегодняшний день являются Vuforia, ARToolKit, WikiTude, LayAR, Kudan AR и другие [4]. Большинство из них являются платными. Среди бесплатных фреймворков выделяют ARToolKit от компании DAQRI и Vuforia от компании Qualcomm. В число плюсов ARToolKit входит поддержка открытого

исходного кода и платформ Android, iOS, Windows, Linux, MacOSX, SGI [5]. Vuforia – AR-фреймворк, позволяющий распознавать 3D-формат и продлевать отображение цели на мобильном устройстве, даже когда она находится вне поля зрения.

Процессы создания виртуальных объектов, написания приложения для считывания маркеров изображения завершаются проверкой и тестированием разработанной системы. Процесс работы дополненной реальности приведен на рисунке.

В рамках исследования возможностей применения AR-технологий в образовании, проводимых на кафедре КПиД Университета ИТМО, планируется разработка обучающего комплекса по английскому языку для детей дошкольного и начального школьного возраста «Happy English». Образовательный проект будет включать несколько обучающих блоков: алфавит, цифры, цвета, местоимения, еда, увлечения и игры, семья и друзья, природа и животный мир, каждый из которых ориентирован на изучение базовых слов и выражений.

Методическая часть проекта будет включать примеры технологических карт уроков, которые будут разрабатываться совместно с квалифицированными учителями английского языка. Техническая сторона проектирования будет включать создание и визуализацию объектов, разработку считывающей изображения программы для смарт-устройств. Учитывая новизну применения AR-технологий в образовании, планируется проведение исследования влияния технологий дополненной реальности на образовательный процесс детей в возрасте от 5 до 9 лет.



Рисунок. Процесс работы дополненной реальности

Дополненная реальность – новая технология интерактивного взаимодействия пользователей с информационными объектами, направленная не только на расширение граней познания, но и на обеспечение новыми технологическими инструментами, позволяющими развивать познавательный процесс учащихся, а также повышать мотивацию к изучению учебных предметов и, в частности, английского языка. Методически целесообразное применение технологий дополненной реальности позволит повысить эффективность обучения и сделать образовательный процесс современным и интересным.

Литература

1. Augmented Reality for Science Education [Электронный ресурс]. – Режим доступа: http://campar.in.tum.de/twiki/pub/Chair/TeachingSs07ArProseminar/1_Display-Systems_Klepper_Report.pdf, своб.
2. VR geek: Новости виртуальной реальности [Электронный ресурс]. – Режим доступа: https://vrgeek.ru/2016/07/21/2467_obrazovanie-v-vr/, своб.
3. Milgram P. Kishino F. A Taxonomy of Mixed Reality Visual Displays. – Munich, 1997. – 388 p.
4. Wagner D., Schmalstieg D. Handheld Augmented Reality Displays. – Austria: Graz University of Technology, 2007. – 197 p.
5. Бойченко И.В., Лежанкин А.В. Дополненная реальность: состояние, проблемы и пути решения // Доклады ТУСУРа. – 2010. – № 1(21). Ч. 2. – С. 161–165.



Шарыпов Павел Андреевич

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4207

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: pavloniym@gmail.com

УДК 004.657

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОСТРАНСТВЕННЫХ ФУНКЦИЙ БАЗ ДАННЫХ

Шарыпов П.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 настоящей работе исследована производительность пространственных функций, реализованных в современных базах данных. Сравнительному анализу подвергались одни из популярных и широко используемых систем управления базами данных – MySQL и MariaDB. Производительность систем управления базами данных тестируется на реальных практических задачах, с использованием большого объема геолокационных данных.

Ключевые слова: Spatial Functions, пространственные функции, геолокация, СУБД, MySQL, MariaDB.

В любом современном web-приложении база данных (БД) является одним из важнейших компонентов, который определяет стабильность и быстродействие системы в целом. Грамотный выбор технологии, которая отвечает современным стандартам и

возлагаемым на нее требованиям, позволяет реализовать качественный и функциональный проект. Основная идея данной работы – сравнить две популярных технологии управления базами данных, и выяснить, в какой из них механизм работы с пространственными данными реализован наиболее эффективно. Работа представляет ценность для разработчиков, создающих комплексные проекты и взаимодействующих с географической и другой пространственной информацией.

Сама по себе БД – это некоторый набор информации, который хранится определенным упорядоченным образом. Взаимодействие с функционалом БД происходит с помощью языка структурированных запросов – SQL, основной задачей которого является предоставление простого способа считывания и записи информации в БД. Помимо стандартных БД, которые позволяют хранить текстовую и числовую информацию, структурированные документы типа XML или, например, JSON-объекты, бинарные и другие файлы, выделяют особый класс – пространственные БД. Такие БД обладают расширенной функциональностью, позволяющей хранить целостный пространственный объект. Такой объект объединяет в себе не только описательную и атрибутивную часть, но и геометрическое представление – данные о положении объекта в пространстве. Традиционные БД имеют возможность использовать индексы, которые повышают производительность и предоставляют более быстрый доступ к необходимым данным. Пространственные БД, в свою очередь, используют специальные пространственные индексы (Spatial Index), позволяющие заметно снизить сложность запросов и количество математических операций о взаимном расположении объектов в пространстве.

Таким образом, функциональность пространственной БД не ограничивается только хранением информации о геометрических объектах, она также оптимизирована для выполнения аналитических запросов, которые содержат специальные операторы для анализа пространственно-логических отношений этих объектов. Такие БД умеют производить поиск расстояний между точками, полигонами и другими геометрическими абстракциями; модифицировать существующие объекты для создания новых (например, разделение полигона на два новых объекта и т.д.); создание новых объектов по координатам; нахождение площади объекта или координаты его центра; и другие вычисления [1].

На рынке современных свободно распространяемых систем управления базами данных (СУБД) существует несколько решений, которые популярны у web-разработчиков:

- MySQL – одна из самых популярных свободных реляционных СУБД, которая используется в тысячах проектах разного масштаба. Данная СУБД имеет определенный функционал для работы с пространственными данными (MySQL Spatial), который постоянно расширяется и улучшается по мере развития проекта;
- MariaDB – другая популярная СУБД, которая является ответвлением от MySQL и разрабатывается сообществом программистов под лицензией на свободное программное обеспечение (GNU GPL). MariaDB также имеет в своем составе функции для хранения и работы с пространственными данными, которые были переписаны и оптимизированы по сравнению с MySQL.

В данной работе будут проанализированы некоторые пространственные функции указанных выше СУБД в рамках конкретных практических задач. В исследовании использован срез данных реального проекта спутникового мониторинга, который включает в себя:

- 35 геометрических объектов – полигонов;
- 4 000 000 пар координат X и Y ;

В ходе анализа использованы следующие пространственные функции:

- центр полигона (ST_Centroid());
- количество точек в каждом полигоне (ST_NumPoints());
- геометрия периметра полигона (ST_ExteriorRing());
- пересечение геометрий полигонов (ST_Overlaps());
- вхождение координаты внутрь периметра полигона (ST_Contains());
- извлечение n -ой точки из периметра полигона (ST_PointN());
- площадь полигона в относительных единицах (ST_Area()).

Каждая СУБД установлена на чистую машину под управлением Ubuntu 14.04. Обе машины имеют идентичную конфигурацию и виртуализируются с помощью Docker с использованием официальных образов. Каждый тест выполнен с использованием флага «SQL_NO_CACHE» и командой FLUSH TABLES, чтобы исключить кэширование запросов в оперативной памяти. Более того, в качестве итогового показателя использовано среднее время выполнения по результатам 10 тестов для каждой исследуемой задачи. Результаты исследования представлены в таблице.

Таблица. Сравнение производительности пространственных функций в исследуемых СУБД

Задача	Время выполнения запроса, с		Разница в скорости выполнения, раз
	MySQL 5.7.17	MariaDB 10.1.20	
Центр каждого полигона	0,00152	0,00036	4,22
Площадь каждого полигона	0,23814	0,13322	1,79
Количество точек в каждом полигоне	0,00044	0,00023	1,91
Проверка на пересечение каждого полигона с каждым	0,20442	0,09920	2,06
Вхождение каждой координаты внутрь периметра каждого полигона (2000000 точек)	44,48506	1,62573	27,36
Вхождение каждой координаты внутрь периметра каждого полигона (4000000 точек)	86,26384	5,49383	15,70

Стоит заметить, что только начиная с версии 5.6, в MySQL реализованы функции не только для минимальных ограничивающих прямоугольников (MBR, minimal bounding rectangles), но и для точных геометрических форм объектов (функции с префиксом «ST_» в названии). В MariaDB такие вычисления доступны с самых первых версий.

К сожалению, ни в MySQL ни в MariaDB не реализована функция вычисления площади полигона на поверхности [2]. Стандартная функция ST_Area() вычисляет площадь только в относительных величинах, поэтому, используя набор других пространственных функций, для вычисления площади на поверхности были написаны две хранимые процедуры.

По результатам тестов можно утверждать, что актуальная версия MySQL уступает по производительности СУБД MariaDB в рамках пространственного функционала. С увеличением объема обрабатываемых данных разница в скорости выполнения увеличивается в десятки раз, что крайне негативно сказывается на быстродействии всего проекта в целом.

В качестве заключения, необходимо отметить, что качество реализуемого проекта напрямую зависит от правильно подобранных технологий. Не стоит забывать, что перед непосредственной реализацией предшествует этап планирования и исследования. В

данной работе была исследована производительность двух популярных СУБД – MySQL и MariaDB. Данные технологии были протестированы на реальных задачах, используя данные с геоинформационного проекта. По результатам исследования можно однозначно определить наиболее предпочтительную СУБД для использования в проектах такого рода.

Стоит заметить, что остается довольно широкое поле для исследований в этой сфере. Необходимо включить в дальнейшие исследования другую популярную СУБД – PostgreSQL, которая имеет большое сообщество разработчиков по всему миру, а ее расширение PostGIS считается одним из лучших для работы с пространственными данными [3]. Также существуют СУБД SQLite с расширением SpatiaLite и MS SQL Server, которые обладают своим пространственным функционалом. Помимо реляционных СУБД существуют и NoSQL решения – MongoDB, которая также работает с геометрическими данными. Все эти технологии имеют свои особенности реализации, обладают своими плюсами и минусами, которые необходимо учитывать при проектировании и разработке проектов такого плана.

Литература

1. Шаши Ш., Санжей Ч. Основы пространственных баз данных. – М.: Кудиц-образ, 2004. – 336 с.
2. MySQL Spatial Functions for Geo-Enabled Applications [Электронный ресурс]. – Режим доступа: <https://www.percona.com/blog/2013/10/21/using-the-new-mysql-spatial-functions-5-6-for-geo-enabled-applications/>, своб.
3. Место PostGIS/PostgreSQL среди СУБД с поддержкой пространственных данных [Электронный ресурс]. – Режим доступа: <http://gis-lab.info/qa/postgis-compare.html>, своб.



Шлемин Сергей Николаевич

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4108

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: JcTorrall@gmail.com



Сокуренко Юрий Андреевич

Год рождения: 1937

Факультет методов и техники управления «Академия ЛИМТУ»,
к.т.н., доцент

e-mail: kpd@limtu.ru

УДК 004

ГЕОЛОКАЦИЯ В МОБИЛЬНЫХ УСТРОЙСТВАХ

Шлемин С.Н.

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 описаны способы геолокации и позиционирования. Подробно разобрано GPS-позиционирование, навигация In-door. Рассмотрено использование данных пользователя для таргетинговой рекламы, а также вопрос их безопасности.

Ключевые слова: геолокация, позиционирование, GPS, альманах, эфимерис, навигация in-door.

Геолокация – определение реального географического местоположения электронного устройства, например, радиопередатчика, сотового телефона или компьютера, подключенного к Интернету. Под геолокацией можно понимать как процесс определения местоположения такого объекта, так и само местоположение, установленное таким способом. Часто для целей геолокации используется та или иная система позиционирования, данные которой преобразовываются в конкретный адрес, географический объект, почтовый ящик, телефон и т.д. [1–3].

Для определения позиции объекта с целью определения его локации, можно использовать ряд базовых методов, таких как:

- счисление координат – способ определения местоположения объекта, по уже известным исходным координатам и параметрам движения. Замер расстояния по сигналу – позиция пользователя определяется по тому, к какой точке идет его сигнал. У каждой точки есть свой идентификатор и радиус действия. Данные о примерном расположении автоматически записываются при приеме сигнала. Анализируя эти данные, данные сигнала, и данные соседних вышек, можно вычислить часто повторяющиеся маршруты. Например, работа, дом, школа и т.д.;
- трилатерация – определение примерной позиции объекта, используя расположения известных двух точек (мобильных вышек), и расстояния между объектом и этими точками. Для увеличения точности анализируется также сигнал, расстояние которого противоположно его силе; измерив время прибытия сигнала, можно тем самым определить расстояние до объекта;
- триангуляция – способ позиционирования, который определяет местоположение измерением углов между объектом и известными точками. Как правило, используются три антенны, также измеряется сигнал для уточнения данных. Данный способ также иногда называют GSM-позиционированием, хотя сегодня любой способ позиционирования мобильного телефона так или иначе использует GSM-сеть, подробнее об этом будет сказано ниже.

Наиболее популярным и эффективным способом позиционирования с последующей геолокацией, является GPS. GPS (система глобального позиционирования) – спутниковая система навигации, обеспечивающая измерение расстояния, времени, и определяющая местоположение во всемирной системе координат WGS 84. Позволяет в любом месте Земли (исключая приполярные области), почти при любой погоде, а также в околоземном космическом пространстве определять местоположение и скорость объектов. Работающими по схожим принципам аналогами данной системы являются Российский Глонасс, Китайский Бейдоу, Европейский Галилео и Индийский IRNSS, последние два не запущены в эксплуатацию. В рамках данной работы была рассмотрена наиболее популярная на сегодняшний день система GPS.

GPS-система состоит из трех основных сегментов. Космический – 32 спутника, которые находятся на средней орбите земли. Управляющий – контролирующие станции управления и мониторинга. Пользовательский – непосредственно приемники GPS-сигнала.

Каждый спутник передает сигналы на нескольких частотах (выделенные L1, L2 и др.). L2 используется в военных целях, L1 открыт для гражданского. Сигналы плохо проходят сквозь большинство твердых объектов, таких как здания и горы. L1 содержит два «псевдослучайных» сигнала, Защищенный (P) код и код гражданского доступа

(С/А). Каждый спутник передает уникальный код, позволяющий GPS-приемнику идентифицировать сигналы. Р-код также называют «P(Y)» или «Y» код. Основной целью этих закодированных сигналов является возможность вычисления времени прохождения (или времени прибытия сигнала) от спутника до GPS-навигатора на Земле. Время прохождения, умноженное на скорость света, равно дальности спутника. Навигационное сообщение содержит данные об орбите спутника, системном времени, общем состоянии системы, а также модель задержки сигналов в ионосфере. Спутниковые сигналы рассчитываются с использованием сверхточных атомных часов.

Навигационные сообщения состоят из:

- телеметрии – используется для синхронизации данных, уникальна для каждого спутника, не меняется в течение долгого времени;
- синхронслова – 30 битное слово, которое содержит сведения о синхронпараметрах, необходимых для отслеживания С/А и Р кодов;
- эфимериса – содержит точные корректировки параметров орбит и часов для каждого спутника, что требуется для точного определения координат. Каждый GPS-спутник передает только данные своего собственного эфимериса. Эти данные действительны только 30 мин. Спутники передают свой эфимерис каждые 30 с;
- альманаха – содержит параметры орбит всех спутников. Каждый спутник передает альманах для всех спутников. Данные альманаха не отличаются большой точностью и действительны несколько месяцев.

Получение данных напрямую со спутников имеет ряд недостатков. Точность – из-за городских и погодных условий, и скорость старта – из-за потери эфимериса и особенно альманаха. В связи с этим сегодня в большинстве мобильных устройств используется AGPS (Assisted GPS). Принцип прост, в случае невозможности получения данных со спутника, телефон берет его из альтернативных источников, например, из GSM-сети. AGPS – самый быстрый и точный способ геолокации на сегодняшний день.

В отличие от навигации на местности, навигация in-door, несколько сложнее, GPS здесь бесполезен, на данный момент существует 2 способа применяемых с переменным успехом:

1. трилатерация на базе Wi-Fi/Bluetooth передатчиков;
2. радиокарта или «цифровые отпечатки» сигналов Wi-Fi\Bluetooth – Wi-Fi/Bluetooth точки, с заранее измеренными значениями, размещаются внутри помещения, местоположение вычисляется путем сравнения измеряемых в реальном времени мощностей сигнала от этих точек.

Вопрос безопасности данных о местоположении клиента предусматривает случаи утечки информации без ведома пользователя, и является огромной проблемой. Речь идет не только о уязвимости приложений к хакерским атакам, но и о злоупотреблении информацией, со стороны владельцев/сотрудников приложения. Популярный сервис Uber не раз критиковался за подобные нарушения. В 2011 г. стало известно, что представитель компании отслеживал все передвижения журналиста, который готовил к публикации статью, критикующую Uber. Представитель компании, который попался на этом, публично извинился и был уволен. Однако в декабре 2016 г. Guardian опубликовал со слов бывшего сотрудника Uber, который утверждал, что безопасность личных данных пользователя не была улучшена с 2011 г., и что ряд работников продолжает следить за VIP-клиентами, возможно для дальнейшей продажи данных об их передвижениях папарацци.

Геолокация используется во многих мобильных приложениях. Дальнейшее улучшение точности позиционирования важно скорее для военной, нежели гражданской сферы. Последним крупным технологическим препятствием остается навигация in-door. На данный момент способы ее реализации не пригодны для широкого использования.

Другой важной проблемой является безопасность данных пользователя, как от внешних атак, так и от злоупотребления ими, в нарушение собственных правил, представителями различных компаний.

Литература

1. Резников М.Б. Геолокация в сотовых сетях с использованием трех базовых станций // Труды научной конференции по радиофизике. – Нижний Новгород: Изд-во НГГУБ, 2005. – С. 202.
2. Wang S., Min J., Yi B.K. Location based services for Mobiles: Technologies and standards. – USA.: LG Electronics Mobile research, 2008. – 122 p.
3. Vukadinovic V., Mangold S. Performance of Collaborative GPS Localization in Pedestrian Ad Hoc Networks [Электронный ресурс]. – Режим доступа: https://s3-us-west-1.amazonaws.com/disneyresearch/wp-content/uploads/20140727175135/p1_mobiopp1.pdf, своб.



Шуклина Алла Сергеевна

Год рождения: 1981

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4206
Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: alla_shuklina@mail.ru



Готская Ирина Борисовна

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна,
д.педагог.н., профессор
e-mail: iringot@mail.ru

УДК 32.019.51

КРИТЕРИИ ЭФФЕКТИВНОСТИ ПРОДВИЖЕНИЯ ПРОЕКТА В СОЦИАЛЬНЫХ СЕТЯХ

Шуклина А.С.

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

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

В настоящей работе рассмотрены критерии эффективности привлечения трафика и внимания к бренду или продукту через социальные платформы – SMM/SMO (Social Media Marketing/Optimization). Обоснованы целесообразность и актуальность использования нефинансовых критериев оценки эффективности SMM. Приведены критерии для оценки эффективности данного канала продвижения.

Ключевые слова: SMM, SMO, маркетинг, веб-технологии, продвижение, социальные медиа.

Продвижение продуктов, услуг, а также брендов компаний в социальных сетях приобретает все большую популярность. Этот тренд обусловлен тем, что социальные медиа способны обеспечивать качественный конверсионный трафик и эффективно

дополнять традиционные каналы его генерации (SEO, PPC) в рамках маркетинговых проектов [1].

Все вышесказанное подтверждает актуальность исследования проблемы оценки эффективности этого способа продвижения.

Ключевая особенность SMM состоит в создании привлекательного для потенциального потребителя контента, который распространяется самими пользователями посредством их социальных связей.

С одной стороны, такая схема способна обеспечить лавинообразный (вирусный) эффект продвижения, так как существующие пользователи привлекают своими действиями новых пользователей, расширяя аудиторию проекта.

С другой стороны, в основе SMM лежит идея создания доверительных отношений с целевой аудиторией, что естественным образом ограничивает публикацию информации рекламного характера, явного предложения товаров или услуг, а также чрезмерно активное продвижение бренда.

Помимо этого, создание сообщества потенциальных клиентов, организация взаимодействия с ними, формирование положительного отношения к объекту продвижения требуют существенных временных затрат. В процессе продвижения необходимо регулярно отслеживать изменение интересов членов сообщества, появление новых трендов.

Ввиду перечисленных особенностей непосредственная оценка окупаемости инвестиций (ROI, см. [2]) в SMM-продвижение возможна далеко не всегда. Поэтому актуальными оказываются нефинансовые критерии оценки эффективности SMM. Их перечень зависит от особенностей объекта продвижения, а также поставленных в рамках SMM-проекта целей. Наиболее распространены следующие критерии:

1. объем трафика, генерируемый SMM;
2. качественные характеристики трафика с сайтов социальных медиа (время пребывания на сайте, глубина его просмотра, действия на страницах и т.д.);
3. показатели конверсии SMM-трафика – доля посетителей ресурса, становящихся клиентами;
4. количественный и качественный состав участников сообществ, организованных в рамках SMM-продвижения;
5. улучшение позиции продвигаемого ресурса в выдаче поисковых систем;
6. число упоминаний, отзывов в интернет (в первую очередь положительных и нейтральных);
7. уровень информированности и доверия потенциальных клиентов.

Первые три из перечисленных критериев могут использоваться для оценки ROI с помощью сервисов аналитики Яндекс.Метрика и Google Analytics. Примеры таких расчетов приводятся в работе [3].

Литература

1. Шалобаев Е.В., Перепелица Ф.А., Шуклина А.С. Исследование и сравнительный анализ продвижения интернет-проектов в социальных сетях [Электронный ресурс]. – Режим доступа: http://research.ifmo.ru/file/stat/194/tom_5_2016.pdf, своб.
2. ВКонтakte [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Окупаемость_инвестиций, своб.
3. FreshITBlog [Электронный ресурс]. – Режим доступа: <https://freshit.net/blog/internet-marketing/kak-podschitat-effektivnost-prodvizheniya-v-socialnyx-setyah/>, своб.

**Шупыра Екатерина Ивановна**

Год рождения: 1991

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4207Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: shupirlik@gmail.com

УДК 004.5

**КОМПАРАТИВНЫЙ АНАЛИЗ МЕТОДОВ ПРОЕКТИРОВАНИЯ
ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ВЕБ-ПРИЛОЖЕНИЙ****Шупыра Е.И.****Научный руководитель – д.педагог.н., профессор Готская И.Б.**

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе автором описаны и проанализированы основные методы проектирования интерфейсов. Рассмотрены этапы процесса проектирования и используемые на этих этапах инструменты.

Ключевые слова: проектирование, пользовательский интерфейс, персонаж, сценарий использования, прототип, тестирование.

Большинство производителей программного обеспечения, и в том числе веб-приложений, все больше уделяет внимания тому, чтобы пользователи чувствовали себя комфортно при работе с их программами и приложениями. Именно поэтому одним из важнейших этапов создания программных средств является проектирование интерфейса.

Существуют несколько подходов к проектированию пользовательских интерфейсов: UCD (User Centered Design), GCD (Goal Scentered Design), ACD (Action Centered Design), TCD (Task Centered Design). Они отличаются тем, что лежат в основе их идеи (цели, задачи или деятельность пользователя), а также критериями качества разработанного интерфейса. Но, в конечном счете, процессы, происходящие на этапах каждого из них, сводятся к изучению пользователя, а точнее к изучению контекста окружающей среды, технических условий, с которыми работают пользователи, сценариев, в рамках которых они обращаются к продукту. В ходе исследования выяснилось, что этапы процесса проектирования, используемые в данных подходах, и их последовательность идентичны, за исключением некоторых формулировок и названий [1].

В проектировании, ориентированном на пользователя можно выделить несколько общих основных этапов:

1. исследование:

- определение пользователей;
- определение пользовательских задач;
- определение требований;

2. разработка прототипа;

3. тестирование.

Каждый из этих этапов может быть адаптирован под существующие требования, свойства и ресурсы проекта [2].

Компании, практикующие подход, ориентированный на пользователя, используют в своей деятельности широкий спектр методов для сбора данных о пользователях и их задачах, анализа потребностей, создания проектных и дизайн-решений, проведения тестирования, оценки удовлетворенности и т.д.

Для решения задач на каждом этапе проектирования существуют собственные методы.

На стадии исследования основными методами сбора информации являются: интервью, опросы, фокус группы, полевые исследования. Одним из результатов полевых исследований и интервью с пользователями является набор поведенческих шаблонов, которые являются основой для разработки персонажей. Персонаж – это совокупность данных об абстрактном пользователе, включающая характеристики пользователя, его опыт, цели и задачи, а также условия окружающей его среды. Профиль персонажа описывает целевого пользователя продукта, дает четкое представление о том, как он вероятнее всего будет использовать продукт и какой результат он ожидает получить.

Для выработки пользовательских и функциональных требований, как правило, необходимо проанализировать сценарии использования. Сценарий (вариант, прецедент) использования – это описание поведения системы, когда она взаимодействует с кем-то (или чем-то) из внешней среды. Система может отвечать на внешние запросы Актера (любой внешней по отношению к системе и взаимодействующей с ней сущности), может сама выступать инициатором взаимодействия», сценарии использования могут быть преобразованы в диаграммы требований, которые описываются с помощью языка UML 2.0 – унифицированного языка моделирования. При использовании GCD проектировщики фокусируются на том, какие вопросы могут возникнуть у пользователя и какие решения ему нужно принимать. В методе GCD целеориентированные сценарии рассматривают не только функциональность системы, но и приоритеты этих функций для пользователя, в то время, как в традиционных вариантах использования все возможные взаимодействия с пользователями считаются одинаково важными и одинаково вероятными [3].

Одна из целей проектирования – получить четкое видение того, каким должен быть интерфейс системы. Основной документ, который получается в итоге этого процесса – детальные схемы страниц (wireframes). Цель таких схем – отразить порядок, структуру, расположение элементов. Таким аспектам дизайна, как цвет и типографика, на этом этапе внимание не уделяется. Они в деталях показывают, какая информация и элементы управления должны выводиться на каждой странице системы, и позволяют расставить акценты. Далее на основе схем создаются действующие прототипы.

Создание прототипа интерфейса – итеративный процесс. Может возникнуть необходимость внести изменения в прототип после тестирования командой или после получения обратной связи от конечных пользователей.

После создания прототипа необходимо оценить насколько качественными являются предложенные решения. Для этого требуется выйти за пределы персонажей и проверочных сценариев и предложить решения реальным пользователям. Тестирование следует проводить тогда, когда решение обрело достаточную детализацию, чтобы пользователи могли реагировать на него как на что-то вполне конкретное, но при этом есть еще достаточно времени, чтобы внести исправления исходя из результатов тестирования. Результатом тестирования, как правило, является доклад о выявленных проблемах и путях их решения [4].

В заключение следует сказать, что наибольшая разница в существующих подходах к проектированию проявляется на первом этапе проектирования. От того, с какой стороны будут рассмотрены пользователи и их задачи, зависит степень эффективности

взаимодействия пользователя с интерфейсом. UCD отличается активным вовлечением пользователей и более детальным изучением контекста использования. Задача GCD понять, почему пользователю необходимо выполнять те или иные задачи. В том случае, когда необходимо выявить ограничения, связанные с особенностями пользователей или неудовлетворенные потребности подходят эти методы [5]. В ACD и TCD акцент направлен на задачи и действия, которые должны быть решены и выполнены с помощью интерфейса, что приводит к более глубокой детализации. Следовательно, их следует применять, если по каким-то причинам сведений о пользователях мало, либо имеется четко определенный набор задач.

Литература

1. User-centered design, activity-centered design, and goal-directed design: a review of three methods for designing web applications // Proceedings of the 27th ACM international conference on Design of communication. – 2009. – P. 1–8.
2. Notes on User Centered Design Process [Электронный ресурс]. – Режим доступа: <https://www.w3.org/WAI/redesign/ucd>, своб.
3. Алан Купер. Об интерфейсе. Основы проектирования взаимодействия. – СПб.: Символ-Плюс, 2009. – 688 с.
4. Сергеев С.Ф., Методы тестирования и оптимизации интерфейсов информационных систем: учебное пособие. – СПб.: НИУ ИТМО, 2013. – 117 с.
5. Солоницын Ю. Task-Centered Design Проектирование, ориентированное на решение стоящих перед пользователями задач [Электронный ресурс]. – Режим доступа: <http://www.slideshare.net/yurysolonitsyn/taskcentered-design-59602663>, своб.



Шустова Ирина Борисовна

Год рождения: 1987

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4108

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: irishiris.insom@gmail.com

УДК 004

ДАнные, ХРАНИМые В ВИде ГРАФОВ. ОБЛАСТИ ПРИМЕНЕНИЯ, ПЕРСПЕКТИВЫ, ПРОБЛЕМЫ МАНИПУЛЯЦИИ

Шустова И.Б.

Научный руководитель – д.т.н., профессор Погорелов В.И.

Работа выполнена в рамках темы НИР № 615892 «Исследования и разработки в области информационных технологий».

В работе рассмотрены перспективы применения графовых баз данных, их отличия от реляционных, недостатки и преимущества перед реляционными базами данных. Рассмотрен вопрос, какие типы информации целесообразно хранить в графовом виде. Ставится вопрос о визуализации.

Ключевые слова: базы данных, графы, хранение информации, SQL, NoSQL, данные.

В течение большого периода развития информационных технологий балом правили реляционные базы данных (БД). Программисты автоматизировали мир. Начав с автоматизации подсчета денег, где табличные записи естественны, практически всю

информацию стали заносить в таблицы. И это работало в течение долгого времени. Из этого мы получили SQL-языка запросов к таблице, простой и логичный язык, которого хватает почти для всех целей.

Но какая информация сегодня на повестке дня? Это уже не просто данные – это весьма непредсказуемая структура, которая со временем может превратиться либо в Big Data, либо в сложную семантическую сеть, и часто разработчик не может заранее предсказать, какой она будет.

Благодаря стандартизации реляционные базы данных заработали себе популярность и доминируют на рынке. Но по факту реляционные БД – это просто таблицы, где в каждой строке выстраивается однозначное соответствие между ключом и его многочисленными (или малочисленными) параметрами. Пока приложения обходились отдельными таблицами и не рассматривали особенных взаимодействий между собой и разными типами данных, этого было вполне достаточно.

Представлять большие и сложные зависимости в таблицах сложно и не эффективно по ресурсам (дисковое пространство, процессор, и главный ресурс – время). Встает вопрос, все ли данные идеально ложатся на табличную структуру? Многие структуры, представляющие практический интерес в математике и информатике, могут быть представлены графами.

Граф [1] – это совокупность непустого множества вершин и наборов пар вершин (связей между вершинами). Объекты представляются как вершины, или узлы графа, а связи – как дуги или ребра. Для разных областей применения виды графов могут различаться направленностью, ограничениями на количество связей и дополнительными данными о вершинах или ребрах.

Граф идеально описывает естественные связи между вещами и цепочки этих связей. Например, структура папок в компьютере или структура организации или генеалогическое древо. Хотя все приведенные примеры можно представить и в реляционном виде, но что если количество узлов превышает тысячу и более? Для этого случая есть специализированный инструмент – графовые базы данных [2].

Итак, реляционная архитектура – это прекрасное решение для тех случаев, когда все просто и однозначно, но совершенно неповоротливая архитектура для создания сложных и гибких запросов, обработки разнообразных и многократных связей между объектами. Однако нельзя забывать о таких преимуществах SQL-баз данных, как возможность создания сложных (JOIN) запросов. Такой подход делает стандартизированные реляционные БД более универсальными, ведь пусть даже большим количеством кода, но каждый запрос может быть в них реализован. Например, найти всех людей моложе 30 лет, у которых есть автомобили синего цвета, будет достаточно легко сделать в SQL, в то время как БД из категории NoSQL потребуют массы усилий для решения этой задачи.

Сегодня ситуация несколько иная. Появившиеся в последние годы так называемые NoSQL (Not only SQL, не только SQL) хранилища реализуют модели данных, имеющие существенные отличия от традиционной реляционной модели. Основная их цель – расширить возможности БД в тех областях, где реляционная модель и SQL недостаточно гибка, и не вытеснять их там, где они справляются со своими задачами. Создатели таких БД среди множества преимуществ использования NoSQL-решений называют высокую производительность при использовании специфических моделей данных и легкость работы с ними. Одним из наиболее популярных и актуальных подвидов нереляционных хранилищ являются графовые БД. Они предлагают более естественное представление информации, основанное на той же логике, с которой люди сталкиваются в реальной жизни. Не секрет, что каждая социальная сеть – это граф, и сетевая модель БД – это тоже фактически граф, но без дополнительных возможностей, которые открывают современную графовую модель, поэтому графовые базы данных

представляют особый интерес для разработчиков.

О графах. Они изначально ориентированы на связи между объектами, и эти связи могут иметь разные характеристики. Например, если требуется разработать БД сериалов, где к каждому эпизоду каждого сериала относятся различные актеры, самым очевидным образом вырисовывается иерархическая модель, которая прекрасно ложится в документарную базу данных. Однако как только появится требование, например, чтобы система также в один клик выводила фильмографию актеров, вся иерархия рассыпается и приходится либо дорабатывать БД, либо менять формат хранения данных. Основным преимуществом графовых БД в этом свете является универсальность, ведь в них можно хранить и реляционные, и документарные и сложные семантические данные. А сама модель построения БД может меняться и модифицироваться в процессе развития приложения без изменения архитектуры и исходных запросов, что значительно экономит время.

С другой стороны, при незначительном количестве связей и больших объемах данных графовые БД демонстрируют более низкую производительность.

Однако в процессе проектирования нередко появляются новые требования к структуре данных, и хорошая модель внезапно может стать плохой. Например, добавление новых связей делает неприемлемой документарную БД, а рост количества JOIN-запросов значительно снижает производительность реляционной БД. В этом случае графы оказываются наиболее универсальным вариантом, позволяющим подстраховаться на случай изменения требований и расширения функционала в будущем. К реляционным данным легко добавить дополнительные связи и без проблем усложнить иерархическую документарную модель [3].

Проекты в области графовых БД начали появляться с конца 1980-х годов, однако в большей степени носили академический характер. В последнее время наблюдается бурный рост интереса к графовым БД в связи с тем, что такая система представления данных оказалась естественной и востребованной в современном мире различных социальных связей (Интернет, социальные сети и т.д.). К достоинствам графовых моделей БД по сравнению с традиционной реляционной моделью исследователи относят не только возможность естественной реализации графовых операций (поиска путей, выделения сообществ и т.п.), но и гибкую схему данных, позволяющую унифицировать хранение разнородных объектов [2].

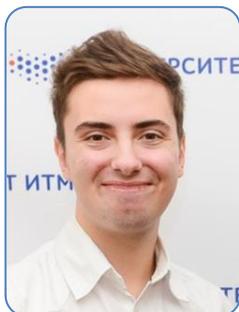
Одновременно с этим возникают специфические манипуляции, применимые только к графовым БД, одна из них – это визуализация [4]. Приведем несколько примеров, когда данные удобнее хранить в графовом виде, и эти данные требуют наглядной визуализации. Это биллинговая информация об абонентах сетей мобильная связи, [5] выявление наиболее популярных направлений и способов связи для оптимизации тарифов; информация и маршрутах клиентов служб такси (вспомним потрясающие визуализации карт городов от Uber); база научных публикаций и связей между ними в форме цитирований и обращений к работам; активность пользователей социальных сетей, выявление популярных пользователей и интересов их аудиторий для размещения рекламы и улучшения работы самих сервисов и многое другое [6–9].

Литература

1. Харари Ф. Теория графов. – М.: Мир, 1973. – 300 с.
2. Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация и применение. – СПб.: БХВ-Петербург, 2003. – 1104 с.
3. Buerli M. The Current State of Graph Databases [Электронный ресурс]. – Режим доступа:

<https://pdfs.semanticscholar.org/5b5b/6b80badccd291e3437460222e24326c65979.pdf>,
своб.

4. Angles R. and Gutierrez C. Survey of graph databasemodel // ACM Comput. Surv. – 2008. – V. 4. – № 1. – С. 1–39.
5. Бартенев М.В., Вишняков И.Э. Использование графовых баз данных в целях оптимизации анализа биллинговой информации [Электронный ресурс]. – Режим доступа: <http://engjournal.ru/catalog/it/hidden/1058.html>, своб.
6. Dominguez-Sal D., Urbon-Bayes P., Gimenez-Vano A., Gomez-Villamor S., Martinez-Bazan N., Larriba-Pey J.L. Survey of graph database performance on the HPC scalable graph analysis benchmark [Электронный ресурс]. – Режим доступа: https://link.springer.com/chapter/10.1007/978-3-642-16720-1_4, своб.
7. Целых А.А., Целых А.Н., Матвеев Д.А. Методы и средства визуализации массивов научно-технических показателей в виде графов [Электронный ресурс]. – Режим доступа: <https://www.science-education.ru/ru/article/view?id=9421>, своб.
8. Апанович З.В. Методы навигации при визуализации графов [Электронный ресурс]. – Режим доступа: <http://www.nsu.ru/xmlui/bitstream/handle/nsu/110/04.pdf?sequence=1&isAllowed=y>, своб.
9. Апанович З.В., Винокуров П.С. Методы и средства визуализации сетей соавторства и сетей цитирования больших научных порталов [Электронный ресурс]. – Режим доступа: <http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2011/part3/AV>, своб.



Щербаков Александр Борисович

Год рождения: 1993

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107

Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: aleksandersher@gmail.com



Готская Ирина Борисовна

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна,

д.педагог.н., профессор

e-mail: iringot@mail.ru

УДК 004

СРАВНИТЕЛЬНЫЙ АНАЛИЗ FRONTEND-ФРЕЙМВОРКОВ СТЕКА РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЙ 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 пяти современных широко используемых фронтенд JavaScript-фреймворков, входящих в стек разработки веб-приложений на платформе Node.js, на основании критериев анализа программного обеспечения: метрик программного обеспечения, показателей валидности исходного кода и результатов тестов производительности.

Ключевые слова: веб-программирование, веб-приложения, JavaScript, фреймворки, фронтенд, метрики программного обеспечения.

Frontend-фреймворки (фреймворки на клиентской стороне) являются неотъемлемой частью стека разработки на платформе Node.js. На их выбор могут повлиять множество факторов, поэтому необходимо провести сравнительный анализ.

Для формальной оценки качества программного обеспечения (ПО) используются метрики – численные меры, которые отражают определенные свойства программного продукта или его спецификации [1]. Авторы исследования [2] использовали ограниченное количество метрик, поскольку некоторые из них отражают одни и те же свойства [3]. Также авторы [2] в качестве критериев использовали результаты валидации кода фреймворков и тестов производительности.

В работе проведен сравнительный анализ производительности, качества и валидности кода пяти наиболее популярных фронтенд JavaScript-фреймворков: React, Angular, Vue, Ember, Backbone. Выбор основан на информации, расположенной на официальных github-репозиториях данных продуктов (табл. 1).

Для теста производительности использованы минифицированные версии исходных кодов фреймворков, в то время как для оценки качества и валидности кода использованы версии для разработки.

Таблица 1. Список фронтенд-фреймворков

Название	Количество «звезд» на github.com	Ссылка на репозиторий
React	54320	https://github.com/facebook/react
Angular	53012	https://github.com/angular/angular.js
Vue	32087	https://github.com/vuejs/vue
Backbone	25687	https://github.com/jashkenas/backbone
Ember	17033	https://github.com/emberjs/ember.js

Метрики были вычислены при помощи прикладного ПО для статического анализа кода Scitools Understand 4.0 (scitools.com), утилит plato (github.com/es-analysis/plato) и complex-report (github.com/philbooth/complexity-report). Результаты подсчета метрик выбранных фреймворков представлены в табл. 2.

Таблица 2. Результаты расчета метрик кода фреймворков

Метрика ПО	Angular	Backbone	Ember	React	Vue
Количество строк кода	28363	1155	25604	2127	6218
Количество выражений	14795	725	12680	873	2653
Количество строк комментариев	5	489	19171	1206	995
Соотношение количества строк комментариев и строк кода	~0	0,42	0,75	0,57	0,16
Цикломатическая сложность	3542	295	3432	414	1172
MI	116	108	113	108	105
Глубина вложенности	9	7	10	8	8

Из результатов значений метрик были получены следующие выводы. Angular почти не содержит комментариев, имеет высокую вложенность блоков – 9 и самую высокую цикломатическую сложность (т.е. количество путей выполнения программы) в своем коде, что сильно усложняет его поддержку. Схожие результаты по

цикломатической сложности и глубине вложенности блоков имеет Ember. В то время как Ember, Backbone и React содержат в своем коде более 50% строк комментариев, что говорит о высокой документированности их исходного кода. Самые низкие показатели цикломатической сложности и вложенности блоков кода имеют Backbone и React, что улучшает их поддержку и отладку проектов, основанных на них.

Тесты валидации были проведены при помощи утилиты JavaScript Lint (jshint.com). В табл. 3 приведены ошибки и предупреждения, найденные при валидации рассматриваемых фреймворков.

Таблица 3. Результаты валидации кода фреймворков

Название фреймворка	Ошибки	Предупреждения
Angular	1	19
Backbone	0	49
Ember	0	1
React	1	101
Vue	1	175

Из результатов валидации были получены следующие выводы. Angular, React и Vue содержат по одной ошибке в их исходном коде, что может привести к некорректному поведению основанных на них приложений. React и Vue содержат самое большое количество предупреждений, однако они не являются критичными. Ember имеет самые высокие показатели валидации кода.

Для проведения тестирования производительности рассматриваемых фреймворков необходимо определиться с предметом оценки. Наиболее корректным будет использование веб-приложений, имеющих абсолютно идентичную функциональность, но реализованных с помощью разных фреймворков и минимальным количеством сторонних библиотек – лишь тех, без которых практическое применение фреймворков невозможно [4]. Под данные критерии подходят веб-приложения todoMVC (todomvc.com), которые представляют собой список дел.

Основным инструментом измерений является инструменты разработчика браузера Chrome. Результаты тестирования производительности представлены в табл. 4.

Таблица 4. Показатели производительности фреймворков

Название фреймворка	Angular	Backbone	Ember	React	Vue
Среднее время загрузки веб-приложения todo, мс	639,85	317,97	898,35	470,09	265,4

Из результатов проведения тестирования производительности были получены следующие выводы. Наилучшие показатели производительности демонстрируют Vue, Backbone и React, в то время как Angular и Ember являются самыми медленными из рассматриваемых фреймворков. Данные результаты напрямую коррелируют с показателями метрик ПО, рассчитанных ранее.

Для выбора наиболее подходящего фреймворка разработчик должен определиться, какой критерий будет иметь большую значимость в рамках конкретного проекта на основании технических требований.

Существенным ограничением проведенного сравнительного анализа является то, что он учитывает лишь формальные «академические» критерии оценки ПО. В то время как на практике выбор фреймворка в большинстве случаев основывается на его популярности, обсуждаемости в профессиональном обществе, полноте документации, частоте выпуска обновлений, архитектурных предпочтениях [4, 5].

Литература

1. Rakic G., Budimac Z. Problems in Systematic Application of Software Metrics and Possible Solution [Электронный ресурс]. – Режим доступа: <https://arxiv.org/ftp/arxiv/papers/1311/1311.3852.pdf>, своб.
2. 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.
3. Barkmann H., Lincke R., Löwe W. Quantitative Evaluation of Software Quality Metrics in Open-Source Projects // Proceedings of International Conference on Advanced Information Networking and Applications. – 2009. – P. 1067–1072.
4. Graziotin D., Abrahamsson P. Making Sense out of a Jungle of JavaScript Frameworks, Towards a Practitioner-friendly Comparative Analysis // Lecture Notes in Computer Science. – 2014. – P. 334–337.
5. Бодров М.Ю. Веб-приложения как эволюционное развитие Web // Сб. научных трудов по материалам III Международной научно-практической конференции «Теоретические и прикладные аспекты современной науки». – 2014. – № 1. – С. 18–21.

**Яковлева Маргарита Александровна**

Год рождения: 1994

Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна, группа № S4107Направление подготовки: 09.04.02 – Информационные системы
и технологии

e-mail: naruyaoi@yandex.ru

**Готская Ирина Борисовна**Факультет методов и техники управления «Академия ЛИМТУ»,
кафедра компьютерного проектирования и дизайна,

д.педагог.н., профессор

e-mail: iringot@mail.ru

УДК 004.4

**ОБЗОР КРОССПЛАТФОРМЕННЫХ ИНСТРУМЕНТОВ РАЗРАБОТКИ
МОБИЛЬНЫХ ПРИЛОЖЕНИЙ****Яковлева М.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».

В работе приведен сравнительный анализ девяти выбранных современных популярных кроссплатформенных инструментов разработки мобильных приложений и их особенностей на основании выявленных критериев.

Ключевые слова: кроссплатформенные мобильные приложения, веб-технологии, фреймворк, разработка мобильных приложений, инструменты разработки.

На сегодняшний день многие пользователи переходят на мобильные платформы, а приложения для мобильных устройств получают все большее развитие. Многие фирмы используют веб-технологии для рекламы своей продукции и услуг. С этой целью создается веб-сайт, а также приложения на популярных операционных системах: Android, iOS и Windows Phone. Создание нативного приложения для каждой из нескольких платформ отнимает много времени у разработчиков, которым приходится учитывать индивидуальные особенности каждой.

Решением данной проблемы является кроссплатформенная разработка приложений. Несмотря на то, что изначально кроссплатформенные инструменты не могли соперничать с инструментами разработки нативных приложений по качеству создаваемого продукта, в последние годы их количество и качество значительно возросло, а потому они требуют внимательного анализа и переосмысления своей позиции на рынке создания мобильных приложений.

Под кроссплатформенным мобильным приложением будем понимать такое мобильное приложение, которое работает более чем в одной операционной системе (или аппаратной платформе).

В работе были рассмотрены следующие инструменты кроссплатформенной разработки мобильных приложений: JUCE (Introjucer), PhoneGap, Xamarin, React Native, AppMobi XDK, Cordova, Appcelerator Titanium, Kony Platform, Telerik Platform.

JUCE (Introjucer) – это открытый кроссплатформенный инструментарий разработки программного обеспечения (ПО) (фреймворк) для языка C++, используемый для разработки GUI-приложений и плагинов. JUCE поддерживает следующие платформы и возможности: Mac OS X (компилируются при помощи Xcode), Windows (MS Visual Studio), Linux, iOS (Xcode), Android (Ant или Eclipse с использованием Android NDK). Имеется бесплатная и платная версии.

PhoneGap – приложение, созданное при помощи PhoneGap, работает как веб-страница внутри WebView, соответственно все строится на основе HTML, CSS и JS. PhoneGap API дает нам возможность использовать все возможности устройства в приложении: камера, звук, GPS, файловая система, контакты, уведомления и т.д. Из-за того, что PhoneGap приложение строится на основе WebView, его можно внедрить в нативное приложение (для разработки гибридного приложения). Приложение можно скомпилировать под любую существующую мобильную платформу, включая Tizen, Bada, Firefox OS. Это открытый и бесплатный продукт.

Xamarin – это фреймворк для кроссплатформенной разработки мобильных приложений (iOS, Android, Windows Phone) с использованием языка C#. Причем, в отличие от Appcelerator, код не интерпретируется на стадии выполнения, а компилируется сразу в нативный код, поэтому поведение, вид и производительность такие же, как и у родных приложений. UI создается для каждой платформы с помощью стандартных для этих платформ View, что увеличивает время разработки, но улучшает качество приложения [1].

React Native – открытый фреймворк от Facebook для кроссплатформенной разработки приложений на языке JavaScript под Android и iOS. Это гибридная система на основе React.js и взаимодействия с родными элементами iOS/Android систем (т.е. с помощью React Native можно не создавать WebView, как это делает PhoneGap, а использовать встроенные компоненты, которые предоставляют вышеуказанные платформы). React Native не использует ни браузер, ни WebView – только JavaScript API поверх нативных компонентов.

Этот фреймворк выполняет JS в отдельном фоновом потоке, взаимодействуя с главным потоком асинхронно, т.е. в потоке JS собирается ряд команд к главному потоку и в определенный момент времени, отправляется сгруппированный запрос (batch-запрос), тем самым никак не блокируя главный поток выполнения программы, в отличие от того же PhoneGap.

Intel XDK является универсальным инструментом для создания мобильных веб-приложений. XDK поддерживает множество игровых сред, в том числе Cocos2d, Phaser, Pixi и EaselJS, предоставляет конструктор приложений типа drag-and-drop, а также различные способы тестирования и отладки приложений. Эмулятор позволяет эмулировать различные функции Cordova API, такие как GeoLocation, Accelerometer и различные события Cordova.

Apache Cordova – это платформа разработки мобильных приложений с открытым исходным кодом. Она позволяет использовать стандартные веб-технологии, такие как HTML5, CSS3 и JavaScript для кроссплатформенной разработки. Приложения выполняются внутри обертки, нацеленной на каждую платформу, и полагаются на стандартные API для доступа к датчикам устройства, данным и состоянию сети [2].

Appcelerator Titanium – приложения, созданные при помощи Titanium, выглядят и ведут себя как нативные, но полностью написаны с использованием JavaScript. UI можно создавать отдельно для каждой платформы с использованием фреймворка Alloy – интегрированный MVC-фреймворк, использующий XML и CSS-подобный синтакс. Несмотря на то, что создание отдельного UI для каждой ОС усложняет разработку и сильно снижает объем переиспользуемого кода, вся бизнес-логика, модель и ядро приложения все же остаются одинаковыми для любой платформы. Titanium позволяет не только разрабатывать приложения, но и тестировать их.

Kony Platform позволяет создавать Web, гибридные и нативные приложения для смартфонов, планшетов и десктопных устройств, написанные с единым кодом. Он поддерживает полный цикл создания приложения (дизайн, разработка, тестирование, развертывание и управление кроссплатформенным приложением), предоставляет гибкую возможность выбирать фреймворки для разработки (включая JavaScript и PhoneGap) и нативные средства (iOS, Android), а также позволяет обновлять приложения, управлять настройками и версиями [3].

Telerik Platform позволяет создавать дизайн, разрабатывать и автоматически тестировать нативные, гибридные и web-приложения. Он предоставляет систему контроля версий и собственную веб-IDE, благодаря чему создавать приложения можно прямо в браузере с любого устройства. Как и PhoneGap, Telerik основан на Apache Cordova со всеми ее достоинствами и недостатками [4, 5].

Проведенный анализ показал, что в настоящее время среди продуктов по разработке приложений представлено большое количество кроссплатформенных инструментов, позволяющих снизить затраты ресурсов на создание ПО. Следовательно, область остается актуальной в настоящее время. Результаты анализа представлены в таблице.

Таблица. Результаты сравнительного анализа

Критерии	JUCE	PhoneGap	Cordova	Telerik	Xamarin	React Native	Intel XDK	Titanium	Kony Platform
Языки программирования	C++	HTML5/CSS и JavaScript			C#	JavaScript	HTML5, CSS, JavaScript	JavaScript	
Платформы	Mac OS X, Windows, Linux, iOS, Android	iOS, Android, webOS, Windows Mobile, Symbian, BlackBerry и Windows Phone 7			iOS, Android, Windows Phone	iOS и Android	Android, iOS и Windows Phone	iOS, Android	
Распространение	платное	бесплатное	бесплатное и платное			бесплатное		бесплатное и платное	
Дополнительное ПО	требуется				нет	требуется			нет

Критерии	JUCE	PhoneGap	Cordova	Telerik	Xamarin	React Native	Intel XDK	Titanium	Kony Platform
Этапы жизненного цикла	разработка		дизайн, разработка, тестирование		разработка, тестирование	разработка	разработка, тестирование		полный цикл
Доступ к нативным элементам UI	нет		нет		нет	есть	нет	да	да

Литература

1. Ермаков О.Ю. Способы кроссплатформенной разработки мобильных приложений // Молодежный научно-технический вестник. – 2015. – № 5. – С. 18.
2. Сосновская А.С. Обзор технологий для разработки приложений для операционной системы Android // Россия сегодня: безопасность, сотрудничество, развитие. Взгляд молодых. – 2016. – С. 236–238.
3. Обзор 7 самых популярных кроссплатформенных мобильных фреймворков [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/229559/>, своб.
4. Жаворонков В.В. Использование фреймворка Xamarin для разработки кроссплатформенных мобильных приложений // Россия сегодня: безопасность, сотрудничество, развитие. Взгляд молодых. – 2016. – С. 175–177.
5. Подробно о Xamarin [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/188130/>, своб.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
Ампилогов Н.С. Типовая структура API социальных сетей.....	4
Ананьева А.А., Шуклин Д.А. Методы тестирования дизайна упаковки.....	6
Анацкая Е.В. Эволюция веб-анимации.....	9
Аринчихина И.В. Исследование кадрового потенциала на автотранспортном предприятии.....	12
Афаунова Д.Р. Символическая система Еmojі как инструмент изучения иностранного языка	14
Барташев А.Н. Сервис для решения задач планирования с использованием диаграммы Ганта на основе системы JIRA	16
Бачурина Л.Р. Анализ технологий генеративного дизайна	19
Башкирцева Е.И. Обзор областей применения компьютерной графики в рекламе	21
Бидянова Е.В. Сравнительный анализ работы библиотек Processing.js, EaselJS, LibCanvas с объектом Canvas.....	24
Бобрик О.А. Перспективные направления развития инфографики.....	27
Богданов Д.Р. Сравнение основных клиентских JavaScript фреймворков.....	30
Букирёва Л.А. Прототипирование как неотъемлемый этап процесса создания сайта.....	32
Ванина Н.В. Цветовая схема web-страницы для людей с нарушением цветовосприятия.....	35
Варганов Д.А. Обзор средств генерации программного кода на основе UML	38
Верхоумов Д.О. Обзор технологий обмена данными в веб-приложении в режиме реального времени	41
Волобуев Н.Г. Сравнение web-фреймворков Python.....	44
Волошин О.В. Методики структуризации эвристических методов в производственных процессах	46
Глебов Р.Г. Исследование процессов защищенности в корпоративных сетях на основе аппаратно-программного комплекса VipNet	49
Головатая Е.Н. Анализ методов брендинга интернет-стартапов	52
Горбунов А.В. Анализ методов рендеринга моделей в системе рендеринга V-Ray для достижения фотореалистичности моделей.....	54
Горбунова А.А. Влияние KPI на мотивацию работников.....	57
Гуляева А.В. Исследование инструментов анализа юзабилити сайтов	60
Гумбург Б.Ю., Гумбург В.Г. Сравнительный анализ антивирусных программ для малого и среднего бизнеса	62
Давлетшин В.Р. Обзор комбинаций фреймворков и других технологий для создания веб-приложений	64
Давыдова А.С., Сокуренок Ю.А. Обзор состояния рынка офисных онлайн-пакетов.....	66
Дерманов М.К. Сравнительный обзор автоматизированных систем для учета задач при разработке веб-проектов	70
Дмитриевская А.А., Мустафина Л.И. Веб-среда как один из вариантов применения картографических интерфейсов.....	74
Довженко М.И. Анализ API управления Bluetooth-устройствами на различных платформах	77
Дорохина О.Ю., Шуклин Д.А. Особенности проектирования дизайна web-интерфейсов с учетом цветового и эмоционального восприятия	80
Дубро А.С. Сравнительный анализ инструментальных средств для кроссплатформенной разработки мобильных приложений.....	83
Жигальцова Е. Особенности инфографики в сети интернет.....	85

Жирнова М.А., Государев И.Б. Применение UML для проектирования и разработки систем на языке JavaScript	88
Жукова А.В. Социально-психологические и моральные методы повышения эффективности системы стимулирования трудовой деятельности персонала в условиях экономической нестабильности	91
Журавлев В.О. Теоретические основы Node.js фреймворков: паттерны асинхронного кода «callback», «coroutine», «promise», «async/await»	94
Захаревич Е.А. Виртуальный тур как наглядный метод презентации проекта	97
Захарова А.О. Сравнительный анализ гейм-дизайна на различных платформах виртуальной реальности	100
Зиборова И.А. Методы и этапы создания дизайна и прототипа пользователь-ориентированных интерфейсов	104
Зуев Е.А. Анализ онлайн-компиляторов и интерпретаторов	106
Ибрагимов Д.И. Анализ современных сервисов для онлайн-вещания мультимедийных презентаций	109
Изукаев Е.В. Исследование внедрения технологий препроцессинга на веб-фронтенде	112
Казначеева Е.О. Эволюция форматов обмена данными на веб-платформе на примере XML и JSON	114
Калашников Д.А. Проектирование и разработка системы для автоматизированного сбора и обработки данных из внешних источников данных	117
Калинина К.В. Выявление начальных ошибок при составлении задания дизайн-проекта	120
Каллауус М.М. Сравнительный анализ инструментов прототипирования интерфейса инженерных систем	122
Катышева П.А., Государев И.Б. Эволюция и проблемы использования микроразметки в вебе	125
Кирикова Ю.В. Методы оценки эргономических показателей пользовательских интерфейсов веб-приложений	128
Костюченко Ю.А. Проблемы веб-семантики в области продвижения веб-ресурсов	131
Красноруцкая Н.С., Перепелица Ф.А., Шалобаев Е.В. Аддитивные технологии для изготовления сложных деталей в машиностроении	134
Кузнецова П.Б. Методика построения 3D-модели объекта на основе данных световых полей	138
Куликова К.Ю. Онлайн-инфографика: типы, цели, особенности дизайна	140
Лисакова А.В. Сравнительный анализ возможностей использования контекстной рекламы поисковых систем Яндекс.Директ и Google Adwords	144
Логинов М.Ю. Исследование современных тенденций в UI/UX	147
Ломтев И.А. Сравнительный анализ веб-систем автоматизированного тестирования задач по программированию	149
Лунев А.В. Сравнительный анализ инструментов дизайна и проектирования интерфейсов web-приложений	152
Магнитова М.В. Анализ сервисов определения позиции веб-ресурса в поисковой выдаче	154
Маревская М.В., Флеров А.В. Инфографика в интернете: онлайн-инструменты	157
Мартюшов М.В., Готская И.Б. Продвижение веб-сайтов: основные методы, проблемы и перспективы	158
Матушевский Н.Е. Определение и анализ недостатков существующих медицинских информационных систем	161

Микитенко А.С., Готская И.Б. Исследование методов и приемов типографики в вебе	164
Митропольская А.М. Модификация программной компоненты Snap Dataset продукта MFE TimeFinder Clone для обеспечения репликации многотомных наборов данных с типом доступа VSAM.....	168
Мустафина Л.И., Дмитриевская А.А. Анимации в веб-дизайне: современное состояние, возможности применения	170
Никифорова А.И. Обзор использования технологий 3D-печати для создания одежды	172
Нифанина А.Р., Шуклин Д.А. Особенности проектирования разверток объемной упаковки сувенирной продукции	175
Павлихин С.В. Использование UML для проектирования JavaScript - приложений	178
Панкратов А.С. Разработка метода повышения видимости сайта в поисковых системах.....	181
Певзнер В.В., Погорелов В.И., Шуклина А.С. Выбор платформы для разработки интерактивного учебника с элементами геймификации.....	184
Передрий О.С., Перепелица Ф.А. Методы разработки на платформе Angularjs V2.....	186
Перепелица Ф.А., Калимуллина А.Ю. Уровни детализации информационной модели на различных этапах жизненного цикла	188
Проводников А.Д. Интерактивные возможности воспроизведения медиаконтента средствами HTML5	191
Пузырев П.К. Сравнительный анализ архитектурных паттернов в iOS разработке	193
Савенкова И.А. Построение игрофицированного приложения на основе структурного подхода.....	196
Сверчков В.А. Обзор фреймворков для отображения картографических данных на мобильных устройствах под управлением iOS	199
Сергейчук Н.А., Готская И.Б. Обзор возможностей современных систем веб-аналитики.....	201
Серебренникова И.О. Электронные образовательные ресурсы для общего образования: современное состояние, проблемы и перспективы	205
Смирнов Д.Ю. Методы оптимизации разработки сайтов с использованием веб-программируемых фреймворков	208
Соболева А.А., Перепелица Ф.А., Шалобаев Е.В. Быстрое прототипирование средствами аддитивных технологий	210
Соловьева Н.Г. Исследование и разработка методов мотивации персонала для улучшения качества работы организации	213
Солодовников А.Ю. Обзор технологии Service Workers (продукта эволюции Web Workers) для оптимизации работы offline-first веб-приложений.....	216
Сохина А.А., Готская И.Б. Обзор популярных систем сборки для фронтенд-проектов.....	219
Ставрова В.С. Исследование технологий оптимизации трехмерных моделей.....	222
Сычева А.В. Психология юзабилити	225
Таскин Н.Е. Исследование методов и средств тестирования web-приложений	228
Тодоровская Я.В. Возможности использования open source продуктов для формирования идентичности компании	231
Топорков П.С. Современные вопросы интеграции CAD и PLM.....	234
Туркунов А.А. Перспективы развития технологии виртуальной реальности в технических системах.....	237
Тхорёва Л.А. Сравнительный анализ онлайн-сервисов для продвижения групп ВКонтакте	239

Фролова М.В. Анализ методов обмена данными при клиент-серверном взаимодействии	242
Хрусталева М.М. Политическая инфографика	244
Чебыкина Д.А. Методы автоматизации подготовки технического задания по выпуску полиграфической продукции	246
Чердынцева Ю.В., Сокуренок Ю.А. Анализ технологий разработки веб-анимации.....	248
Черепанов Д.М. Технологии трехмерного моделирования предметов интерьера и мебели	251
Чуканбаева Н.Г., Перепелица Ф.А. Проблематика сохранения целостности данных информационной модели при ее экспортировании из среды разработки в среду просмотра	254
Шадрин Д.М. Анализ возможностей семантической верстки веб-сайтов для лиц с ограниченными возможностями.....	258
Шакирзянова Е.И. Технологии дополненной реальности как инструмент повышения качества образовательного процесса детей дошкольного и начального школьного возраста	261
Шарыпов П.А. Сравнительный анализ пространственных функций баз данных	264
Шлемин С.Н. Геолокация в мобильных устройствах	267
Шуклина А.С. Критерии эффективности продвижения проекта в социальных сетях.....	270
Шупыра Е.И. Компаративный анализ методов проектирования пользовательских интерфейсов веб-приложений	272
Шустова И.Б. Данные, хранимые в виде графов. Области применения, перспективы, проблемы манипуляции	274
Щербаков А.Б. Сравнительный анализ Frontend-фреймворков стека разработки веб-приложений на платформе node.js	277
Яковлева М.А., Готская И.Б. Обзор кроссплатформенных инструментов разработки мобильных приложений	280

**АЛЬМАНАХ НАУЧНЫХ РАБОТ
МОЛОДЫХ УЧЕНЫХ УНИВЕРСИТЕТА ИТМО
Том 4**

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 24.11.2017

Заказ № 4056

Тираж 100 экз.