

ТЕХНИЧЕСКИЕ НАУКИ*Сафронов Михаил Алексеевич*

магистрант

НИУ ИТМО

г. Санкт–Петербург

Зингеренко Юрий Александрович

технический директор

НПП Новел–ИЛ

г. Санкт–Петербург

**ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНТЕРНЕТ–ПРИЛОЖЕНИЙ
С РАСШИРЕННЫМИ ВОЗМОЖНОСТЯМИ.****RIA – DESIGN AND DEVELOPMENT**

Аннотация: в статье автор описывает подходы к проектированию и разработке интернет–приложений с расширенными возможностями (англ. RIA – Rich Internet Application).

Ключевые слова: интернет–приложение, проектирование интернет–приложения, разработка интернет–приложения, пользовательский интерфейс, бизнес–логика, RIA, Java, GWT, Vaadin.

Что общего между обычным выключателем и диммером? Они являются интерфейсами – с их помощью можно управлять освещением. Обычное интернет–приложением и RIA тоже интерфейсы – они позволяют пользователю взаимодействовать с контентом через web–сервер.

Тогда зачем менять очередную технологию, если главная задача достигнута? Ответ в различиях – вторые варианты предоставляют больше возможностей управления.

В RIA рабочая область радует красивыми всплывающими окнами или раскрывающим списком, реагирующим на наведение курсора или правой кнопки мыши – больше не нужно обновлять всю страницу ради одного сообщения. А на одной странице можно разместить контент разного типа с точностью до пикселя, сохраняя при этом межплатформенную совместимость.

Далее в статье рассматриваются общие вопросы проектирования пользовательского интерфейса (далее – UI) и бизнес–логики (далее – BL) RIA, а также выбор фреймворка для его последующей разработки.

Интернет–приложения можно разделить две основные группы: информационно–аналитические (напр. еgr–система, социальная сеть) и коммерческие (напр. сайт–визитка, интернет–магазин).

Информационно–аналитическое приложение обладает сложной ролевой структурой – у целевых пользователей (далее – ЦП) общие задачи, но разный уровень доступа и взаимодействия с контентом (содержимым страницы). Таким образом, главной задачей BL–проектирования является определение ролей, отражающее главные задачи приложения, построение между ролями логических взаимосвязей, и нормализация данных в соответствии с принятой логикой. В задачу UI–проектирования входит разметка рабочих страниц, страницы пользователя, новостной страницы.

Целью коммерческого приложения является продажа услуги или товара ЦП. В этом случае задача BL–проектирования сводится к анализу текущих и будущих предпочтений ЦП в сравнении с конкурентными предложениями в сопоставимом сегменте. На основании такого обзора подбираются те медиа, которые определяют поведение ЦП сделать целевое действие – звонок или заказ через форму. В задачу UI–проектирования входит разметка посадочных и контентных страниц, форм заказа. В отличие от информационного приложения, где контент наполняется самим ЦП, в коммерческом приложении он создается разработчиком.

Рабочей документацией BL–проектирования может быть одна или несколько диаграмм, входящих в состав UML (англ. Unified Modeling Language), например, диаграмма классов, последовательности или вариантов использования. Рабочей документацией UI–проектирования являются макеты разнотипных страниц в векторном или растровом виде и сопроводительная CSS –документация (англ. Cascading Style Sheets) по верстке.

Неправильное определение ключевых сценариев и неспособность выявить долгосрочные последствия могут поставить под угрозу все приложение. Следующие рекомендации относятся к любому RIA:

– Учитывать, на каких платформах пользователь будет взаимодействовать с приложением. Как правило, уровень представления информационно–аналитических приложений с настольных компьютеров расширяется до мобильных приложений, а коммерческих – до версии для мобильного устройства (планшет, смартфон) и, иногда, мобильного приложения. Уменьшение размеров экрана (4–10) и отсутствие мыши у ЦП могут значительно влиять на графический дизайн.

– Видимая производительность имеет намного большее значение, чем фактическая. Поэтому решающее значение имеют управление ожиданиями и знание типовых схем взаимодействия с пользователем. В некоторых ситуациях, даже очень короткая задержка в доли секунды, может создать ощущение, что приложение не отвечает.

– Если информационно–аналитические приложения ориентированы на спокойный графический дизайн и классический UI, то коммерческие приложения могут быть выполнены без таких ограничений.

Разработка RIA может проводиться на различных технологиях. Сам термин «RIA» впервые был упомянут компанией Macromedia в официальном сообщении от марта 2002 года. Изначально он подразумевал только технологию Flash, позднее – Silverlight и JavaFX, а развитие JavaScript привело к его расширению на другие платформы.

До появления этих технологий, обычное интернет–приложение реализовывало парадигму Thin client, в которой отображение контента осуществлялось через стандартный HTML. Существующая параллельно парадигма Fat client открывала больше возможностей для взаимодействия, но требовала предварительной установки десктопного приложения.

К недостаткам первой технологии относился низкий уровень UI–функциональности и использование парадигмы Page flow, а ко второй – сложность в развертывании и последующей поддержке при растущем количестве ЦП.

В течение этого периода глобальные линии связи увеличили пропускную способность и надежность соединений – количество активных интернет–пользователей растет и браузер становится самым популярными интерфейсом. Коммерческие проекты, стремясь увеличить рентабельность бизнеса, повышают интерактивность взаимодействия с ЦП за счет упрощения UI и насыщения медиа. Старые технологий дополняются новыми – это приводит к необходимости стандартизации, главной целью которой является ускорение процесса разработки приложений и повышения их надежности.

Одна из таких объединяющих технологий – AJAX. Она использует программный интерфейс DOM (англ. Document Object Model), с помощью которого браузерный движок обрабатывает полученные данные, доступные пользователю, в результате чего их представление изменяется. CSS обеспечивают согласованный внешний вид элементов приложения и упрощает обращение к DOM–объектам, а объекты XMLHttpRequest (или подобные механизмы) используется для асинхронного взаимодействия с сервером, обработки запросов пользователя и загрузки в процессе работы необходимых данных. Принцип её работы становится одним из архитектурных шаблонов, входящим в состав таких популярные фреймворков как GWT или Vaadin.

В рамках дипломной работы в качестве технологии разработки выбран фреймворк Vaadin по следующим причинам:

– Готовый каркас, использующей событийно–ориентированную архитектуру. Это позволяет сконцентрироваться только на BL и UI, оставив вопросы безопасности и производительности разработчикам Oracle, а совместимости – разработчикам Vaadin.

– Широкий спектр виджетов (клиентских компонентов), отвечающих последним требованиям UI, как в стандартной библиотеке, так и написанными сторонними авторами.

– Vaadin TouchKit для разработки мобильных приложений под операционные системы IOS и Android.

– Java в качестве единого язык программирования. Такое преимущество достигнуто за счет использования Java–сервлетов на стороне сервера, и GWT – на стороне клиента.

В задачи этапа разработки входит перевод UML–диаграмм в используемый язык программирования, выбор виджетов (активных графических элементов), соответствующих требованиям UI–проектирования, и верстка пользовательского интерфейса по CSS.

Рабочей документацией на этом этапе является скомпилированный и проверенный на работоспособность код в составе WAR–файла.

Заключительными этапами разработки является установка на web–сервер заказчика необходимого программного обеспечения (напр. среды исполнения, реляционной базы данных), развертывание и тестирование приложения.

Подробное описание каждого этапа представлено в дипломной работе, где в качестве проекта рассматривается ерп–система под графические представления настольного компьютера и мобильного приложения. В качестве веб–контейнера используется сервер приложений GlassFish, развернутый на удаленном VPS.

Список литературы

1. Nicolas Fränkel. Learning Vaadin. /Nicolas Fränkel. Packt Publishing. – 2011. – С. 7–25.
2. Marko Gronroos. Book of Vaadin. / Marko Gronroos. Vaadin Ltd. – 2013. – С.57–67, С. 405–473, С. 677–724.
3. Крэг Ларман. Применение UML 2.0 и шаблонов проектирования. / Крэг Ларман. – 2013. – С. 277–294.
4. Ном. Проектирование программного обеспечения. / Ном. Тематические Медиа. – 2009. – <http://habrahabr.ru/post/74330/>.
5. SECL GROUP. Блог компании. / SECL GROUP. Тематические Медиа. – 2013. – http://habrahabr.ru/company/SECL_GROUP/blog/.
6. Авторская группа. Руководство Microsoft(R) по проектированию архитектуры приложений. / <http://microsoft.com/architectureguide/>. – 2009. – С. 21–67, С. 149–173.