УДК 004.921

С.П. Некрашевич, Е.А. Магдайчук
Донецкий национальный технический университет, г. Донецк
кафедра программного обеспечения интеллектуальных систем

РАЗРАБОТКА АВТОНОМНОГО ВЕБ-ПРИЛОЖЕНИЯ "MULTIMEDIA МАТЕМАТИКА"

Аннотация

Некрашевич С.П., Магдайчук Е.А. Разработка автономного веб-приложения "Multimedia Математика". *Проанализировано функционирование веб-приложений. Проведен анализ обеспечения автономности клиента. Предложен подход к адаптации. Рассмотрены составляющие одностраничных приложений. Описаны архитектура и ресурсы разработанного приложения.*

Ключевые слова: *мультимедиа, веб-приложение, автономное приложение, HTML5, JavaScript, SPA, одностраничное приложение, AngularJS, JSON, SVG, MathML.*

Постановка проблемы. Электронный учебник "Multimedia Математика", разработанный в ДонНТУ [1], является педагогическим программным средством по дифференциальному, интегральному исчислению и векторной алгебре. Учебник содержит теоретический материал с формулами, графическими и видео-иллюстрациями, звуковыми комментариями, интерактивными задачами. Приложение свободно распространяется как на DVD-дисках, так и файловым архивом для установки на ПК под управлением ОС Windows.

Перенос материала учебника в Интернет с реализацией в виде платформенно-независимого веб-приложения является актуальной задачей. Функционирование такого приложения на мобильных устройствах, в частности, смартфонах, планшетах, электронных книгах, требует адаптации с учетом обеспечения автономности клиентской части приложения при отсутствии постоянного соединения с сервером.

Цель статьи – провести анализ обеспечения автономности клиента в модели веб-приложения и описать архитектуру разработанного приложения.

Постановка задачи исследования. Классическая архитектура веб-приложения обладает характерной логикой работы с ресурсами в сети Интернет:

1) идентификация разнородных ресурсов по URI;

2) HTTP-протокол транспорта ресурсов и взаимодействия с сервером;

3) отображение презентационной логики ресурсов на основе HTML.

Необходимо адаптировать данную логику для обеспечения автономности клиента, выполняемого в браузере мобильного устройства, а также

реализовать средства отображения мультимедиа-информации с учетом предложений, изложенных в [2].

Решение задач и результаты исследований. Необходимость постоянного соединения накладывает ограничения на процесс взаимодействия клиента с сервером – веб-приложение без Интернета существовать не может.

Приложения Rich Internet Applications (RIA), построенные на технологиях JavaFX, Flash/Flex, Silverlight, частично решают вопрос автономности, но не поддерживаются на некоторых мобильных устройствах. Они передают веб-клиенту необходимую часть пользовательского интерфейса, оставляя большую часть данных на сервере.

С появлением HTML5 появляется возможность создания автономных (offline) веб-приложений и постоянного хранения данных на стороне клиента. Программное обеспечение клиента получает средства повторного использования ресурсов локально без обращения к серверу.

Страница HTML используется в качестве контейнера для ресурсов, определяя DOM-иерархию элементов графического интерфейса пользователя (GUI). При этом каскадные таблицы стилей CSS3 позволяют гибко конфигурировать и изменять стили элементов в реализации динамичного и реалистичного GUI.

Язык JavaScript, выполняемый в каждом браузере, является связующим звеном среди ресурсов и позволяет построить на нём всю функциональность автономного приложения. Использование специализированных JavaScript библиотек-оберток вокруг элементов DOM, например jQuery, позволяет подавлять перегрузку страниц при гиперпереходах и активации форм, производить AJAX-взаимодействие с сервером, пр. В результате страницы не перегружаются целиком, а догружают необходимые данные асинхронно с сервера и внедряют в DOM, что делает веб-приложение интерактивным и производительным.

Нотификация JSON также является частью формата представления данных JavaScript, занимая место формата XML в AJAX-взаимодействии с сервером для представления запросов, ответов и самих ресурсов.

В настоящее время приобрели популярность т.н. одностраничные веб-приложения, Single Page Applications (SPA) [3], реализованные на JavaScript. Это приложения, которые загружают одну HTML-страницу и динамически обновляют ее при взаимодействии с пользователем без обращения к серверу.

Одним из средств реализации одностраничных приложений является фреймворк AngularJS [4], который предлагает структурированную среду для разработки динамических веб-приложений. Этот программный каркас использует собственный шаблонный язык для внедрения зависимостей в GUI, реализует шаблон проектирования MVC в разделении логики данных и отображения, производит двустороннее автоматическое связывание данных.

Ресурсы веб-приложения. Архитектура автономного веб-приложения “Multimedia Математика” спроектирована в виде одностраничного

приложения на основе фреймворка AngularJS, реализующего SPA.

Для выбора и загрузки приложения реализована серверная часть на языке JavaScript с помощью программной платформы Node.js и MongoDB в качестве базы данных, которая хранит весь материал учебника, результаты тестирования пользователей, статистику работы с системой, пр.

Материал учебника представлен в специально разработанном формате на основе JSON, который содержит представление всех ресурсов, как информационных, так и интерфейсных. Этот формат не зависит от платформы и языка реализации и может быть использован в других проектах в качестве презентационного базиса.

Работа пользователя с приложением происходит во взаимодействии с браузером, в котором выполняется автономная клиентская часть под управлением JavaScript. Пакеты ресурсов загружается с сервера при первичном обращении и во время последующих сеансов синхронизации. Манифест (конфигурационный файл) пакета, описывающий состав пакета, представляется в формате JSON. Вводится понятие внутреннего URI, которое применимо к ресурсам, находящимся в пакете.

Таким образом, весь материал электронного учебника представлен в декларативном виде, и, в отличие от HTML, возможна гибкая конфигурация элементов GUI и подстройка под каждого пользователя “на лету”, также упрощается интеграция с последующими расширениями приложения.

Транспорт Спроектирован адаптер использования протокола HTTP, который выполняет поддержку транспорта ресурсов при множестве асинхронных запросов 2 типов:

- обращение к диспетчеру клиента;
- обращение к серверу при синхронизации.

Обращение к объекту диспетчера производится для обновления визуальной части графического интерфейса – смена страниц, запуск анимаций и звуковых комментариев, проведение тестов. При этом переходы, нажатия кнопок, обработка форм, которые в классической модели приводят к транспорту на сервер, подавляются.

Обращение к серверу производится посредством AJAX-запроса для получения пакетов ресурсов и сохранения статистики работы с приложением. Синхронизация может производиться фоново и бесшовно с учетом текущего прогресса в изучении материала учебника, осуществляется как при появлении соединения, так и по расписанию.

Презентационная логика. Диспетчер клиента обрабатывает внутренние запросы для построения GUI на клиенте, производя создание элементов и внедрение их в DOM-иерархию динамически на основании описания ресурсов с использованием библиотеки jQuery и работы специализированного шаблонизатора. При этом генерация результирующего HTML-файла не производится.

Графический материал иллюстраций и анимаций представлен в векторном формате SVG, который позволяет выполнить масштабирования без потерь и поддерживается большинством браузеров. Традиционные пиксельные графические форматы файлов не используются. Вопрос полноценного использования анимаций остается открытым.

Формулы и их фрагменты, встречающиеся в материале, представлены в формате XML MathML, их код вставляется непосредственно в текст и визуализация производится непосредственно браузером.

Для поддержки мультиязычности ресурсы содержат текст с локализацией на различных языках, что делает возможным переключение интерфейса и материала учебника, “на лету”. Реализованные языки – украинский, русский, английский, добавление нового языка тривиально.

Отображение материала производится покадрово и в виде прокручиваемой ленты. Наличие оглавления, индекса, закладок, поиска, системы гиперссылок и вызова контекстных подсказок делают навигацию в приложении удобной и интуитивно-понятной.

Выводы. Проанализированы особенности функционирования веб-приложения, проведен анализ обеспечения автономности клиента. В результате предложен подход к адаптации на основе одностраничного приложения. Описана архитектура реализации автономного веб-приложения с использованием библиотеки AngularJS, обладающего следующими достоинствами:

- не требует носителя или установки;
- не требует постоянного Интернет-подключения;
- может использоваться как сервис.

Реализованное автономное веб-приложение может использоваться как средство самостоятельного изучения и контроля знаний по высшей математике, а также в дистанционном образовании.

Список литературы

1. Электронное пособие "Multimedia Математика" / Интернет-ресурс. – Режим доступа: www/ URL: http://pois.donntu.edu.ua/uk/nauchnaya-rabota/elektronnye-uchebniki/elektronnoe-posobie-multimedia-matematika.html

2. Некрашевич С.П. Розробка інтерактивних засобів навчання та контролю знань на основі електронних книг / С.П. Некрашевич / II міжнародна науково-практична конференція “Сучасні інформаційні системи і технології” – Суми, СНТУ 2013. – сс.85-86.

3. Michael S. Mikowski, Josh C. Powell Single page web applications. / M. Mikowski – Planning Shelter, Island, 2013. – 433c.

4. Pawel Kozlowski, Peter B. Darwin Mastering web application development with AngularJS: build single-page web applications using the power of AngularJS. / P. Kozlowski – Birmingham, 2013. – 372c.