

РАЗРАБОТКА ИНТЕРАКТИВНЫХ ЭЛЕКТРОННЫХ КНИГ ФОРМАТОВ EPUB 3 И HTML 5

КУРОЧКИН А. Е.

*Белорусский государственный университет информатики и радиоэлектроники
(г. Минск, Республика Беларусь)*

E-mail: kurochkin@bsuir.by

Аннотация. Обсуждаются технологии разработки электронных книг с интерактивным контентом и пути решения проблемы безопасного обмена данными. Представлены результаты разработки курса лекций по дисциплине “Радиоприёмные устройства” в виде электронной книги на основе формата EPUB 3 и в виде локального веб-сайта.

Abstract. Technologies for developing e-books with interactive content and ways to solve the problem of secure data exchange are discussed. The results of the development of a course of lectures on the discipline “Radio receivers” in the form of an electronic book based on the EPUB 3 format and in the form of a local website are presented.

Введение

Как известно, компания Adobe, разработавшая популярный плагин Flash Player для проигрывания мультимедийных файлов в браузерах, объявила о полном прекращении поддержки плагина и платформы для создания контента для него с 2021 года [1]. Компания предупреждала об этом решении ещё в 2017 году. Частично это связано с недостатками этой технологии в виде огромного количества уязвимостей в системе безопасности как сайтов, так и браузеров. Все последние версии браузеров, в частности Google Chrome, в последние годы открыто запрещают использование Flash, мотивируя это в первую очередь соображениями безопасности, а последние версии операционных систем Windows 10 и 11 и вовсе просто блокируют работу всех опасных плагинов, в том числе и Flash. Таким образом технология Flash официально окончательно закрыта.

Казалось бы, какое отношение всё сказанное имеет к разработке электронных книг? Если считать, что контент книги – это только текст и картинки, то никакого. Но это только на первый взгляд. Все так называемые ридеры или читалки электронных книг представляют конечному пользователю контент только после анализа данных произвольного типа или гипертекста, описанных с помощью языков разметки (markup) XML (eXtensible Markup Language), HTML (Hyper Text Markup Language) или XHTML (eXtensible Hypertext Markup Language). Так называемое тело всех документов такого типа состоит из специальных заранее предопределённых элементов разметки – тегов и непосредственно содержимого документа – данных. Именно теги показывают читающему текст программному обеспечению как представлять содержащиеся в документе данные на странице. Как правило ридеры чаще всего основаны на обычных браузерных движках для отображения содержимого книг – т.е. фактически это интернет-браузеры. Современный же подход к созданию электронных книг заключается во внедрении в контент книг интерактивных визуальных элементов, а таковыми до недавнего времени как раз и были Flash-анимации.

Анализ текущей ситуации

В течение ряда последних лет автор разрабатывал конспекты курса лекций в виде электронных книг на основе наиболее востребованного формата EPUB (Electronic Publication) версии 3 [2], что связано с расширенной поддержкой использования новых форматов шрифтов, аудио– и видео, нелатинских языков, языка разметки формул MathML и языка программирования JavaScript. Во всех разработанных электронных версиях курса лекций для интерактивных анимаций различного рода как раз и применялась вышеупомянутая технология Flash как форма мультимедийного программного обеспечения. Был хорошо отработан достаточно простой способ создания интерактивностей на основе анимированных слайдов PowerPoint с последующей конвертацией в Flash с помощью программного продукта компании iSpring [3]. Но ситуация изменилась и после прекращения поддержки плагина Flash Player функциональность и полезность разработанных электронных книг существенно пострадала. Необходим переход на альтернативную технологию создания интерактивного визуального контента.

HTML 5 (Hyper Text Markup Language, version 5 [4]), язык разметки гипертекста, впервые выпущенный в 2008 г., серьёзно обновился в 2014 г. Целью обновления было улучшение языка и расширение его возможностей, чтобы соответствовать растущим требованиям разнообразного мультимедийного веб–контента. Правда сам по себе HTML 5 нельзя использовать для анимации или интерактивности, его необходимо дополнять, например JavaScript – другой перспективной веб–технологией. HTML5 сразу стал одним из фаворитов в области создания интерактивного веб–контента. Он универсален и работает во всех современных браузерах без особых проблем в сфере интернет–безопасности.

HTML 5 добавил несколько полезных API (Application Programming Interface) для новых и имеющихся элементов, нацеленных на улучшение разработки веб–приложений и предназначенных для устранения недостатков предыдущих версий HTML. Один из таких API специально создан для работы с аудио и видео через использование тегов <audio> и <video>. Этот API предоставляет возможность проигрывания аудио и видео и устраняет необходимость использовать сторонние приложения, такие как Flash, для отображения медиа. Тег <canvas> может быть использован для отрисовки графики через скрипты (обычно используется JavaScript) и выполнения интерактивных анимаций. Следует отметить, что компания iSpring уже с 2012 года также обеспечивает конвертацию анимированных слайдов PowerPoint в HTML 5. Но это было ещё до вышеупомянутого решения компании Adobe.

Казалось бы, что всё складывается удачно и все технологии разработки электронных книг готовы перейти на новый уровень. Но, всё оказалось не так просто. Как ни странно, но причиной этому стала опять отмеченная выше безопасность. Дело в том, что в целях безопасности обмена информацией ещё в 1995 г. разработан CORS (Cross Origin Resource Sharing [5]) или механизм совместного использования ресурсов между разными источниками – механизм, использующий дополнительные HTTP (HyperText Transfer Protocol) заголовки, которые определяют: дать браузеру разрешение на доступ к ресурсам на другом источнике (домене), отличном от текущего, или нет. Говорят, что браузер пользователя делает запрос с другого источника, если источник текущего документа отличается от запрашиваемого ресурса доменом. Т.е. в целях безопасности браузеры ограничивают cross–origin запросы, инициируемые скриптами. Это значит, что веб–приложения могут запрашивать HTTP–ресурсы только с того домена, с которого были загружены, если только не используются CORS–заголовки, разрешающие на загрузку другого домена для безопасного использования.

В нашем случае потенциально опасным оказался тег <image>. При попытке рисовать на холсте (canvas) любые данные, которые загружаются из другого источника без одобрения CORS, холст становится потенциально опасным (tainted). Вызов методов getImageData(), toBlob(), toDataURL() на canvas на таком холсте приводит к возникновению

сообщения системы безопасности SecurityError следующего содержания: “Access to image from origin has been blocked by CORS policy: Cross origin requests are only supported for protocol schemes: http, data, isolated-app, chrome-extension, chrome, https, chrome-untrusted”. Это означает, что чтение файлов даже с локального диска, например, по схеме адресации: file:/// [имя диска]:/[путь к файлу на диске]/[имя файла], тоже приведёт к ошибке, связанной с CORS.

В программных EPUB 3 ридерах, в частности, в ридере Azardi от компании **Infogrid Pacific**, основанном на движке от веб-браузера Mozilla, проблема уже CORS решена. В результате разработка электронной книги в формате EPUB 3 свелась к простой замене всех интерактивных Flash-сюжетов в конспекте лекций на альтернативную анимацию с помощью JavaScript и HTML5. Но при разработке электронного варианта конспекта лекций в HTML 5 работа некоторых скриптов приводила к ошибке. В частности, проблема с CORS возникла при написании и отладке интерактивного демонстрационного учебного скрипта на языке JavaScript, который осуществлял сжатие изображения с потерями по стандарту JPEG. Исходное изображение было размещено в HTML элемент canvas.

Конечно, самым радикальным способом обойти все проблемы с CORS – это запуск браузера, например Chrome, с отключённой защитой. Для этого следует указать дополнительные ключи либо с помощью командной строки, либо с помощью ярлыка на рабочем столе: "C:\Program Files\Google\Chrome\Application\chrome.exe" --disable-web-security --user-data-dir=" Chrome dev session". В этом случае Chrome скопирует себя во временную папку "c://Chrome dev session" и позволит для одной сессии запустить требуемые “опасные” файлы для “местного” безопасного просмотра. Папку “c://Chrome dev session” в последствии можно будет удалить, если она не будет нужна. Такой вариант вполне допустим в случае разработки электронной книги на основе локального веб-сайта, хотя и требует дополнительных условий для запуска браузера. Но было решено найти альтернативный путь решения проблемы.

Подсказка есть в самом сообщении об ошибке, где указаны допустимые протоколы обмена данными. В основном они сводятся к необходимости создания локального веб-сервера для формирования необходимых HTTP заголовков, что тоже явно не в интересах пользователя электронной книги. Но вот протокол data [6] позволяет решить описываемую проблему. В схеме протокола data данные произвольного вида представлены последовательностью кодов с использованием кодировки ASCII диапазона безопасных символов для адреса URL (Uniform Resource Locator), который выдан ресурсу в интернете и использование стандартной шестнадцатеричной кодировки для символов за пределами этого диапазона. По состоянию на 2022 г., эта схема полностью поддерживаются большинством основных браузеров. Наиболее предпочтительным является способ кодирования изображений в стандарте Base64 [7]. Base64 – это стандарт кодирования двоичных данных при помощи 64 символов ASCII. Алфавит кодирования содержит латинские символы A–Z, a–z, цифры 0–9 (всего 62 знака) и 2 дополнительных символа “+” и “/”. Максимальный размер URL по протоколу data для браузера Google Chrome, например, составляет 2 Мбайта [8], что вполне достаточно для кодирования изображений приемлемого качества.

Было принято решение использовать для загрузки в canvas не локальный файл изображения, а его версию, закодированную в Base64. Так как в этом случае данные об изображении представляет собой ASCII символы, то это делает возможным встраивание закодированного изображения непосредственно в тело скрипта, а проблема CORS в этом случае не проявляется. Объём файла исходного BMP изображения 320 на 320 пикселей составил примерно 300 Кб, а длина Base64 адреса – 409 Кб.

Заключение

Таким образом, использование технологий HTML и JavaScript позволило разработать два варианта конспекта курса лекций по дисциплине “Радиоприёмные устройства”: в виде электронной книги на основе формата EPUB 3 и в виде локального веб-сайта, для просмотра которого не требуется локальный веб-сервер. В обоих вариантах для создания интерактивного контента применены технологии HTML 5 и JavaScript без Flash и проблем, связанных с CORS. Полученные результаты использованы при разработке ЭОР – электронного обучающего ресурса по дисциплине “Радиоприёмные устройства”.

Список использованных источников

1. Adobe Flash Player EOL General Information Page [Электронный ресурс]. – Режим доступа : <https://www.adobe.com/products/flashplayer/end-of-life.html>
2. EPUB 3.2 [Электронный ресурс]. – Режим доступа : <https://www.w3.org/publishing/epub3/epub-spec.html>
3. iSpring [Электронный ресурс]. – Режим доступа : <https://www.ispring.ru/company>
4. HTML5 [Электронный ресурс]. – Режим доступа : <https://www.w3.org/TR/2011/WD-html5-20110405/>
5. Same-origin policy [Электронный ресурс]. – Режим доступа : <https://www.securitydrops.com/same-origin-policy/>
6. The "data" URL scheme [Электронный ресурс]. – Режим доступа : <https://www.rfc-editor.org/rfc/rfc2397>
7. Base64 [Электронный ресурс]. – Режим доступа : <https://www.rfc-editor.org/rfc/rfc4648>
8. Excellent Export and the Chrome URL limit [Электронный ресурс]. – Режим доступа: <https://craignicol.wordpress.com/2016/07/19/excellent-export-and-the-chrome-url-limit/>

УДК 004.354.3:159.93-056.24