

УДК 004.942

## ИСПОЛЬЗОВАНИЕ WEB-ТЕХНОЛОГИЙ В ФОРМИРОВАНИИ АУДИОВИЗУАЛЬНОГО КОНТЕНТА НА ВНУТРЕННЕМ ТЕЛЕВИДЕНИИ ВУЗА: ПРЕИМУЩЕСТВА И НЕДОСТАТКИ

**Филиппов Дмитрий Сергеевич,**  
*аспирант кафедры вычислительных систем и сетей,*  
*e-mail: dsfilippov@ui-miit.ru,*  
*Российский университет транспорта (МИИТ),*  
*г. Москва*

*Статья посвящена обобщению опыта управления web-контентом, который используется в качестве аудиовизуального источника на внутреннем телевидении вуза. Предлагается методика визуализации контента, позволяющая автоматизировать процесс вывода информации с корпоративного сайта вуза на внутренние ТВ-экраны и «интерактивный киоск», а также избежать избыточного дублирования контента. Рассматриваются возможности открытых стандартов для web-технологий, которые предлагаются в качестве отсутствующей альтернативы для систем Digital Signage. Рассматриваются результаты апробации предложенной методики использования web-технологий в формировании аудиовизуального контента на внутреннем телевидении вуза. Формулируются методические рекомендации по оптимизации и реализации автоматизированного вывода информационного контента на внутреннем телевидении вуза (Indoor TV) и возможности использования единого источника контента и единой системы управления web-контентом, объединяющей сайт и Indoor TV вуза.*

**Ключевые слова:** внутреннее телевидение, сеть цифровых дисплеев, каскадные таблицы стилей, система управления контентом, открытые web стандарты, Smart TV, «интернет вещей»

## THE USAGE OF WEB-BASED TECHNOLOGIES FOR AUDIOVISUAL CONTENT OUTPUT ON THE UNIVERSITY'S INDOOR TV – THE ADVANTAGES AND DISADVANTAGES

**Filippov D.S.,**  
*postgraduate student at the department computer systems and networks,*  
*e-mail: dsfilippov@ui-miit.ru,*  
*of the Federal State Institution of Higher Education “Russian University of Transport” (MIIT)*

*The article is devoted to the generalization of web-content management experience, which is used as an audio-visual source on the internal television of the University. The method of content visualization is proposed, which allows automating the process of information output from the corporate website of the University to the internal TV screens and “in-active kiosk”, as well as avoiding excessive duplication of content. The possibilities of open standards for web-technologies, which are offered as a non-existent alternative for Digital Signage systems, are considered. The results of testing the proposed method of using web-technologies in the formation of audiovisual content on the internal television of the world are considered. Methodical recommendations on optimization and implementation of automated output of information content on the internal television of the University (In-door TV) and the possibility of using a single source of content and a single web-content management system that combines the website and Indoor TV of the University are formulated.*

**Keywords:** Indoor TV, Digital Signage, cascading style sheets, content management system, open web standards, Smart TV, “Internet of things”

DOI 10.21777/2500-2112-2018-2-78-86

Свою историю web-браузеры начинали как программы визуализации гипертекстовой разметки. Со временем функции в браузерах развились до того, чтобы эмулировать нативные (native) приложения, создаваемые под какую-то конкретную аппаратную платформу. В настоящее время HTML5 API, наряду с AJAX открывают новые возможности по отображению контента на больших ТВ экранах, всё чаще размещаемых в учебных заведениях России и других стран [1, 2]. Современные тенденции развития информационно-коммуникационных технологий (ИКТ) заставляют производителей встраивать браузеры в сами телевизоры, называя такой подход Smart TV. В будущем это поможет открыть сферу внутреннего телевидения (ТВ) по-новому, избавившись от подключённого к экрану компьютера (аппаратного плеера в виде неттопа) и оставив только «облачный» сервер, на котором будет храниться контент и управляющая информация. Это «интернет вещей» в чистом виде.

подавляющее большинство вузов имеют свои интернет-сайты. Объём контента на таких сайтах, как правило, прямо пропорционален количеству студентов [5]. Базовая идея проектируемой модели формирования аудиовизуального контента на внутреннем телевидении вуза заключается в создании единой системы управления контентом на сайте вуза, экранах внутреннего ТВ и сенсорном интерактивном киоске. Это позволяет использовать большое количество готового и обновляемого web-контента с сайта на экранах, не дублируя его, не привлекая новых специалистов и не задействуя для этого стороннее специализированное программное обеспечение. Технологии вывода изображения аудиовизуального контента и получения обратной связи могут быть совершенно разными – от воспроизведения с DVD-диска до потока IPTV (Internet Protocol Television). Поэтому необходимо учитывать, что предложенная модель может быть не эффективной в других сферах применения или оказаться ресурсозатратной для некоторой конфигурации ИКТ вуза.

Под понятие телевидение больше подходит определение классического потокового IPTV. Сценарии использования ТВ непосредственно в вузе весьма обширны – от анонсов предстоящих мероприятий (собраний, круглых столов, конференций и т.п.) до трансляций обучающего материала. Помимо классического ТВ (Indoor TV), телевидение вуза включает сеть Digital Signage (видеореклама на мониторах) и интерактивные системы (например, сенсорные информационные киоски). Digital Signage (DS) – это решение для централизованного управления и распространения разнородного контента на удаленных системах распространения (цифровые панели, видеостены, мониторы). DS представляет полноценный инструмент коммуникаций со студентами, сотрудниками и посетителями вуза, предлагающий в своей основе механизмы, позволяющие оптимальным образом доносить динамически меняющийся контент до аудитории. Ключевое достоинство DS систем – их универсальность и гибкость, позволяющая демонстрировать на дисплеях самую разнообразную информацию, а также взаимодействовать с ней. На DS можно добавить возможность просмотра расписания (группы, преподавателя), систему записи на курсы дополнительного профессионального образования, заказ справки из кадровой службы, систему записи на приём к руководству и многое другое. При этом демонстрируемый контент может варьироваться от статичного текста и изображений до видео высокого разрешения со звуковым сопровождением.

Классическая инфраструктура Digital Signage включает такие компоненты, как плееры, вещательные серверы и сети передачи данных. Зачастую распределение контента от источника к устройствам отображения по витой паре привычно воспринимается как сетевое решение с передачей по стеку TCP/IP, хотя на самом деле здесь возможно множество различных вариантов [3, 4]. В частности, может использоваться сеть передачи данных с использованием кабеля с витыми парами как физической среды, но при этом передаваемый сигнал может быть закодирован закрытыми проприетарными алгоритмами. В этом случае на передающем устройстве сигнал преобразуется из формата HDMI (High Definition Multimedia Interface) в формат, разработанный производителем устройства, и передаётся по кабелю. На принимающей стороне приёмник того же производителя конвертирует полученный сигнал обратно в HDMI – формат и передаёт его в телевизор. Идея такого подхода заключается в том, что удалённый ТВ «думает», что работает с HDMI сигналом напрямую. Другое преимущество такой сети в ее низкой стоимости при достаточно большой дальности передачи данных: возможна доставка видео на расстояния до 300 м, но при увеличении дальности поддерживаемое разрешение видео, конечно, необходимо снижать. Следует отметить, что самой современной на текущий момент является технология

HDBase-T. Эта технология позволяет помимо несжатого видео высокого разрешения по той же «витой паре» передавать другие сигналы, включая аудио, кадры Ethernet со скоростью 100 Мбит/с, управляющие сигналы RS-232 и инфракрасные сигналы, а также подключать порт USB 2.0 и обеспечивать питание устройств с потребляемой мощностью до 100 Вт. Но, как и в первом случае технология HDBase-T также функционирует по каналам «точка–точка» и через стандартные коммутаторы Ethernet «не передается». HDBase-T является открытым стандартом, разрабатываемым одноимённым альянсом во главе с такими компаниями как LG Electronics, Samsung Electronics, Sony Pictures Entertainment [4]. По сравнению с закрытыми технологиями, в данном случае нет привязанности к конкретному производителю приёмников и передатчиков.

Если на передающей стороне взять HTML за основу вывода информации на экраны ТВ, то при таком подходе достаточно через серверный язык программирования (например, PHP) подключиться к базе данных имеющегося сайта, чтобы вывести необходимый контент в браузер ТВ или в браузер компьютера (подключённого к ТВ по выше описанному методу) [8]. Рост популярности HTML5 как современного web-стандарта изменил внешний вид интернет сайтов. HTML пятой версии – идеальная платформа для создания веб-приложений с использованием медиа. Чтобы подключиться к уже имеющейся базе данных, нужен аппаратный или программный «плеер». В идеале этим «плеером» должен быть простой браузер, что было реализовано в качестве эксперимента в ходе данного исследования.

Сейчас в России не малое количество компаний занимаются разработкой программных и аппаратных плееров для рынка Digital Signage. Производители аппаратных медиаплееров указывают поддержку HTML5 в характеристиках. Однако поддерживать визуализацию современной HTML разметки могут далеко не все плееры. Другими словами, медиаплеер не сможет безошибочно воспроизвести созданный где-либо ещё, кроме встроенного «конструктора», сложный HTML контент. Во многих медиаплеерах встречается устарелое программное обеспечение, не позволяющее воспроизводить контент, созданный более новыми способами программирования. Сложность использования HTML5 в системах Digital Signage заключается в том, что каждый медиаплеер – как отдельный браузер: со своим уникальным способом воспроизведения контента и с закрытой «экосистемой». Обновить любой браузер гораздо проще, чем сделать прошивку в аппаратном плеере. Однако такие «закрытые» плееры не позволят подключиться ни к одной системе управления сайтом и соответственно не смогут подгружать или синхронизировать информацию с ней (особенно если речь идёт о «самописной» системе сайта, а не распространённом «движке»). Тут нужно понимать отличие web-контента (созданного с использованием HTML/JS/CSS) от системы управления контентом (Content Management System – CMS), которая должна быть единой с сайтом (или иметь лёгкие механизмы синхронизации баз данных и необходимых сопроводительных медиа-файлов). Если речь идёт о сайте, то CMS использует стек web-технологий (как минимум server-side язык программирования для взаимодействия с СУБД, например, PHP+MySQL). Но если речь идет о перечисленных «закрытых» медиаплеерах, то в большинстве случаев в них защита программа, которая не имеет к web-технологиям никакого отношения – она лишь умеет воспроизводить простой гипертекст, вставленный в её редактор вручную (каждый медиаплеер делает это по-своему) [3].

Развитию индустрии Digital Signage препятствует отсутствие единого формата соответствующих систем. Продукты Digital Signage представляют собой закрытые системы. Размещение контента (не имеет значения гипертекста или нет) в нескольких сетях Digital Signage, построенных на базе разных продуктов, связано с техническими трудностями, каких не наблюдается на телевидении и в Интернете. Отсутствие общего коммуникационного протокола затрудняет работу с несколькими сетями, повышает стоимость и затрудняет расширение самих сетей. Управление таким проектом представляет собой непростую задачу.

После анализа недостатков нативных проприетарных CMS рассмотрим преимущества системы, полностью реализуемой на web-технологиях (как client-side, так и server-side).

**1. Открытый стандарт WEB.** Помимо открытого исходного кода web-разметки, существует организация, разрабатывающая и внедряющая технологические стандарты для Всемирной паутины – World Wide Web Consortium (W3C). Эти стандарты подразумевают больше возможностей для доработки и индивидуализации интерфейсов. Открытые технологии, знакомые множеству разработчиков,

позволяют создать такие решения, которые не предлагает ни один коробочный программный продукт в данной области.

**2. Портитрование шаблонов** (структуры для вывода контента). Ещё одним немаловажным плюсом является лёгкий перенос шаблонов вывода из одной CMS в другую. HTML/XML структура для вывода контента из единой базы данных (БД) всегда остаётся неизменной. В разных CMS, конечно, могут быть отличия в шаблонизаторах, но как правило, они не столь радикальные как функции «Импорт-Экспорт» в классических Digital Signage, которые, в лучшем случае, смогут перенести структуру шаблона на другой компьютер с CMS того же производителя (о совместимости с другими системами речи быть не может). Нативные же клиенты могут открывать только то, что создано в их «экосистеме».

**3. Кроссплатформенность.** Для web-среды основной платформой является браузер и протокол HTTP с базовым набором методов GET и POST. Протокол передачи гипертекста стандартизирован, его легко отладить в случае возникновения каких-то проблем. Благодаря кроссплатформенности запуск web-контента осуществляется без установки какого-либо программного плеера. Браузер присутствует в любой системе, начиная с Windows, заканчивая Android.

**4. Масштабирование.** Систему, построенную на web-технологиях можно как развернуть, так и нарастить на порядок быстрее классической DS. На конечном узле достаточно открыть URL того ресурса, с которого происходит «вещание» web-контента. Основное, но не ключевое требование – наличие интернета (ниже в статье рассмотрен вариант, когда Интернета нет или он нестабилен). Несмотря на использование web-технологий и протокола HTTP в качестве доставки контента на экраны ТВ, доступ к сети Интернет не является обязательным условием. В плюсы масштабирования можно отнести и наличие «облачного» (а не только локального) сервера для вещания и управления контентом, т.е. реализация модели SaaS. Основопологающим будет являться наличие «движка» (CMS) сайта вуза. Для подключения новых сотрудников вуза (для добавления контента) нет необходимости устанавливать какие-либо клиентские программы работы с сервером, т.к. с ним можно будет работать в браузере, соответственно и нет ограничения на количество пользователей и их локацию, этого нельзя сказать о проприетарных программах, где клиентское ПО обязательно должно быть установлено на всех машинах, с которых осуществляется управление/редактирование контента. Нет необходимости обновления каждого клиентского компьютера, как, к примеру, при выходе обновления клиентской нативной программы его разработчиком.

**5. Функциональность.** Это свойство оценивается неоднозначно. Например, трудно отдать предпочтение, если сравнивать гибкость нативной программы Microsoft PowerPoint и его web-аналога Prezi.com, позволяющего, используя всю мощь JavaScript и CSS, создавать анимированные презентации ничуть не хуже. Кроме того, Prezi.com можно вставить в любой шаблон Digital Signage, основанный на web-технологиях. Аналогичными могут быть Adobe Edge и WIX.com – обе программы созданы в качестве визуальных конструкторов web-контента с довольно солидным функционалом каждая, но первая работает под Windows в качестве родной программы, а вторая использует «прослойку» в виде браузера для доступа к своим сервисам. То же самое можно сказать про классику – Microsoft Word и WYSIWYG редакторы для web-систем (взять, к примеру, TinyMCE). Конечно, набрать текст в известном редакторе и вывести его на стенд специалисту того же учебного отдела проще, чем осваивать аналогичный web-based инструмент. Но с другой стороны, может быть менее трудоемким один раз занести эту информацию в систему, из которой при необходимости затем выводить на разные цифровые информационные системы. В данном случае каждый потребитель ориентируется на тот функционал, который нужен именно ему. Но возвращаясь к открытости web-стандартов, всегда проще найти разработчика для реализации необходимого функционала под стандартный web-интерфейс, чем под малораспространённый продукт.

К неоднозначным моментам можно отнести и внедрение интерактивности. Одно дело – просто организовать вещание с некоего источника на экран на стене, а другое дело – спроектировать удобный и красивый интерфейс для взаимодействия с пользователями через сенсорный экран. В пользу более богатого функционала под web-платформу говорит наличие в JavaScript такого инструмента как TouchEvent для обработки событий сенсорных экранов [7].

**6. Адаптивность.** Это набирающий популярность тренд последних лет, который уже произвёл революцию в создании сайтов. Смысл его в том, что один и тот же web-контент может выглядеть совер-

шенно по-разному на разных экранах. Можно запрограммировать шаблон вывода так, что на ТВ экране с вертикальной ориентацией он будет помещать тот же контент, что и на экране с горизонтальной ориентацией, не прибегая к его масштабированию (хотя это возможно автоматически) и не допуская пустых мест. Технически это будет один и тот же шаблон и контент. Раньше, поскольку технологию адаптивной верстки не поддерживали браузеры, для этих целей создавали разные версии сайта (например, на поддомене pda.\*.\*). Сейчас это реализуется благодаря таким неотъемлемым web-инструментам как каскадные таблицы стилей (CSS и @media запросы). Благодаря этим таблицам достигается независимость от разрешения экрана. Например, два ТВ экрана с разным разрешением Full HD (1920x1080) и Ultra HD (4K, 3840x2160) могут одинаково гармонично выводить web-контент (даже фиксированной ширины, например, видео), если использовать специально разработанный для этого шаблон. На 4K ТВ он может не растягиваться на всю ширину, а занимать только половину или 2/3, а на оставшейся части выводить что-то ещё, например, загружать слайдер с изображениями более высокого качества (в последнем случае не обойтись без HTML5 тега <picture>). В классическом Digital Signage для этого пришлось бы создавать отдельные шаблоны вывода под каждую ориентацию экрана и его разрешение [1].

**7. Множественное использование web-контента.** Существенное преимущество, которое не сможет предложить нативная Digital Signage система под какую-либо конкретную операционную систему. Это своеобразное множественное наследование, повторное использование уже ранее созданного web-контента без необходимости его дублирования на других экранах/шаблонах. Единственная версия этого контента будет храниться только в одном месте и нигде больше (возможен, конечно, вариант с синхронизацией, но он должен быть автоматизирован) и будет подгружаться на те экраны, где она нужна (при необходимости с разным стилевым оформлением). Если изменится хоть одно слово в этом тексте, изменение произойдёт и на всех экранах (частях экранов), подгружающих из БД этот фрагмент текста. При этом внести изменение достаточно только в одном месте, например, в системе управления контентом сайта вуза. Соответственно на сайте изменение произойдёт мгновенно (вариант с кэшированием не рассматривается), а на экранах в зависимости от частоты обновлений, предусмотренной шаблоном вывода либо управляющими данными с сервера. Если в тексте встречается какое-то слово, строка или ссылка, которую нужно исключить с экранов ТВ (но оставить на сайте вуза), то это легко решается через то же CSS. Один и тот же класс стилей будет иметь разные свойства – display:inline-block на сайте и display:none на ТВ.

К неявным на первый взгляд плюсам можно отнести отсутствие привязки лицензии к каждой машине (оплата в таких системах может взиматься, например, за количество пользователей, если взять за основу классические SaaS модели), а также одновременная работа с системой неограниченного количества участников-операторов. При использовании классических программ с контентом на сервере может работать, как правило, только один оператор.

Итак, были рассмотрены преимущества Frontend интерфейсов систем Digital Signage, основанных на стеке web-технологий.

Перед рассмотрением недостатков, сделаю оговорку – если мы допускаем наличие аппаратного медиаплеера с операционной системой Windows или Linux (или его производными, например, Android) и т.п. (а не только браузер, встроенный в ТВ с технологией Smart TV), то большую часть этих минусов можно нивелировать. Далее отмечаются недостатки системы, полностью реализуемой на web-технологиях.

**1. Производительность.** Специализированные приложения разрабатываются под ту или иную операционную систему и всегда впереди стандартных браузеров при тех же задачах. Особенно это заметно при воспроизведении видео в высоком (например, 4K) разрешении или сложных анимационных переходах между слайдами (HTML блоками), сделанными на HTML5 Canvas или WebGL. Одним из подходов к решению этой проблемы является использование аппаратного плеера с поддержкой нужной производительности в зависимости от разрешения экрана и необходимых визуальных операций. Другой вариант – использование CSS3 анимации (Cascading Style Sheets, каскадной таблицы стилей), адаптированной для создания плавной анимации и переключения элементов интерфейса, если не требуется какое-то сложное решение. Анимации на CSS3 будут выполняться несколько быстрее аналогичных, реализованных на языке JavaScript (и его производной библиотеке jQuery), т.к. браузеру заранее извест-

но конечное состояние элемента и сама функция анимации [6]. Однако в CSS3 анимации нет полного контроля над каждым кадром.

**2. Человеческие ресурсы.** До тех пор, пока полностью построенные на базе web-технологий решения не появятся на рынке в качестве коробочных продуктов, при эксплуатации DS системы будут требоваться специалисты с HTML+JS+CSS знаниями. Такие продукты не появятся быстро, т.к. они хороши при индивидуальном проектировании системы и не позволят первое время создать универсальное решение, на которое нацелены извлекающие прибыль DS компании. Если проектируется full stack web-based система, то понадобится специалист и по backend платформе со знанием серверных скриптовых языков программирования (PHP/JSP/Perl/ASP.NET/ Node.js). Примеры WYSIWYG редакторов HTML разметки имеются, однако они заточены для создания сайтов и, к сожалению, они адаптируются под конкретную серверную backend платформу (часто без предоставления её открытого кода). Необходимость в server-side разработчиках имеется, если планируется внедрение индивидуальных решений. Несмотря на то, что в коммерческих native системах всегда есть конструкторы контента, сделать что-то индивидуальное на проприетарной платформе не представляется возможным ввиду закрытого кода «движка». В традициях российского web-программирования часто бывает, что web-разработчик является по совместительству также дизайнером, художником и собственно программистом. Вполне ожидаемо, что нанятый web-разработчик не будет обладать опытом создания контента для сферы Digital Signage ввиду довольно молодого тренда на российском рынке. Следует отметить, что построить простой web-интерфейс может любой желающий без специальных знаний. Существует много решений для создания HTML5 контента. В качестве примера – Adobe Edge, входящий в состав Creative Cloud – инструмента от компании Adobe, разработанного для создания web-контента. Adobe Edge позволяет быстро подготовить впечатляющий контент, но в качестве «начинки» для нативных Digital Signage систем с ним могут возникнуть трудности. Дело в том, что Adobe Edge был разработан для современных браузеров, а, как уже было отмечено, не все медиаплееры могут гарантировать поддержку новых технологий HTML5 (если проигрывание HTML контента происходит не в браузере, а в специальном нативном плеере).

**3. Невозможность взаимодействия с системой.** По сути это означает, что нет никакой связи web-приложения с уровнем ниже браузера. Современные браузеры в целях безопасности даже ограничивают чтение локальных файлов. Взаимодействовать с операционной системой можно через плагин Java для браузеров (Java апплеты) или через технологию ActiveX (которая работает только в браузерах компании Microsoft). Можно попытаться решить какие-то задачи через другой плагин – Adobe Flash Player, но он также имеет огромное количество ограничений безопасности и в 2018 году практически «сошёл на нет». Не используя перечисленные выше методы, невозможно даже считать системное имя компьютера, не говоря уже об изменении какой-нибудь ветки реестра для смены системных настроек. Кроме того, метод HTML5 FileReader в этом случае бесполезен, так как пользователю в браузере необходимо вручную выбрать нужный для чтения файл (или парсинга, если речь от текстовом/XML файле, смотря для чего мы его открываем в браузере), что для создания полной автоматизации является недопустимым решением. Единственное, что можно сделать «на лету», считать системные дату и время. Любое взаимодействие с драйверами подключённых устройств или с портами ввода-вывода, недопустимо, ну и само собой выключить экран на ночь также не получится.

**4. Сложности в создании расписания вещания.** Этот пункт является следствием предыдущего. Все протестированные нативные клиенты для создания контента и расписания работают по следующему принципу: создание «экрана» (слайда), создание плейлиста(ов) из «экранов», создание расписания запуска плейлистов. Если никаких проблем с первыми двумя этапами не возникает, то для создания полноценной сетки вещания (особенно, если «экранов»/слайдов с разнородным контентом несколько), использования одних лишь web-технологий недостаточно. На сегодняшний день проблема с расписанием вещания web-контента на распределительном компьютере-плеере решается запуском разных URL с помощью программы для автоматизации xStarter. На данный момент на рынке полноценных рабочих full-stack web-инструментов для автоматизации вещания и мгновенных push-обновлений web-контента нет. Прототипом для создания таких «браузерных» систем могут служить «Progressive Web Apps» (PWA): HTML5 Web Worker, WebSocket, Node.js и AJAX/JSON.

**5. Зависимость от Интернета.** Хотя классические нативные DS системы также могут зависеть от доступности Интернета (обновить погоду, RSS-ленты, новости аккаунтов ВКонтакте/Twitter/Facebook и т.д.), в аналогичных web-решениях такая зависимость выше, т.к. необходимо загружать весь контент и даже саму структуру «экрана». Недоступность web-based информационных DS систем при offline режиме можно решить двумя методами: развернув контент-сервер непосредственно в стенах образовательного учреждения и реализовав решение HTML5 Service Worker (сделав специальную offline версию). Также разработчики web-системы могут добавить возможность кэширования и повторное использование соединений для тестирования сети перед обновлением «экрана». Это позволит отображать HTML5 контент независимо от состояния сети. Правда в случае необходимости использовать один и тот же контент в интернет среде (например, на сайте образовательного учреждения) и на локальных системах, придётся искать решение для автоматизации синхронизации двух баз данных. В целом, зависимость web-системы от Интернета хоть и минус, но не критический, т.к. один «экран» с разными источниками контента может быть разделён на два-три-четыре... независимых благодаря использованию плавающих фреймов `<iframe>` или `jQuery .load/.ajax` и независимому обновлению друг от друга без обновления всего окна браузера. Если один из «экранов» даст сбой, то это не сильно испортит общую картину.

**6. Требования к видео контенту.** Несколько лет назад это действительно был весьма существенный недостаток. Он начал проявляться после перехода популярного сервиса YouTube.com с технологии Adobe Flash на HTML5. Разные браузеры могли воспроизводить ограниченное количество видео форматов (кодеков) и для тега `<video>` требовалось прописывать сразу несколько источников в разном формате, чтобы проблем с воспроизведением не было. Одним словом, была зависимость от кодеков, встроенных в браузер. На сегодняшний день существует такой формат видео, который одинаково хорошо воспроизводят все без исключения браузеры – это \*.mp4/h.264 (контейнер mp4 с кодеком h.265/hevc на данный момент воспроизводится только в последней версии браузера Safari). Если требуется открытие и проигрывание видео в разных Digital Signage браузерных системах, то никаких проблем с воспроизведением mp4/h.264 не будет (кроме общих описанных выше проблем при воспроизведении видео в высоком разрешении). Естественно, никаких проблем с автостартом или отложенным стартом видео при загрузке web-«экрана» нет, – всё решается через простейший скрипт на JavaScript.

**7. Расхождения в отображении контента.** Всегда могут быть совсем незначительные расхождения в отображении web-«экрана» в зависимости от браузера. И чем более новые web-инструменты используются, тем выше вероятность этого расхождения потому, что тот или иной браузер может быть не совместим с используемой технологией. Едва ли это может быть недостатком, т.к. и в нативных DS «плеерах» могут быть аналогичные проблемы с совместимостью. Самый распространённый пример из другой области программного обеспечения, но который актуален и на сегодняшний день – Microsoft Word 2003 и ниже не сможет открыть документ (\*.docx), сохранённый без обратной совместимости со старыми версиями (\*.doc). В отличие от классических сайтов, в web-based системах может иметь смысл привязки Frontend контента к какому-то конкретному браузеру (включая случай, когда речь идёт о встроенном в сам ТВ браузер). В сфере образовательного учреждения это имеет смысл, т.к. web-контент не транслируется за пределы корпуса вуза, а в его стенах работает одно и тоже программное обеспечение. Полное решение проблемы небольшого расхождения в отображении контента в разных браузерах – это использование HTML тегов и CSS стилей, рекомендованных W3C, а также избегание экспериментальных web-технологий, нерегламентированных CSS селекторов атрибутов, браузерных префиксов CSS свойств и т.д. Так или иначе, качество вёрстки web-«экрана» – это философский момент и расхождения блоков на пару пикселей или немного иное сглаживание шрифтов совершенно не скажутся на общей картине.

В качестве вывода следует отметить, что несмотря на описанные недостатки, использование web-based системы DS в сфере образования оправдано и позволяет интегрировать контент для ТВ в БД сайта, образуя единую систему управления контентом, чего очень не хватает в современных нативных DS комплексах. Не хватает и единого стандарта. Такие системы перестанут частично или полностью работать в случае, если изменится структура сайта, его вёрстка или API (если сайт отвечает на такие запросы). В случае web-based системы, – и Frontend и Backend составляют единую неразделяемую платформу. Запросы на обновление контента к серверу (к тому, где расположен сайт, если это более

приоритетный источник обновления) возможны из встроенного в Smart TV браузера без использования какого-либо программного и аппаратного обеспечения.

В проводимом в рамках исследования эксперименте видео сигнал (с запущенным web-«экраном») от одного компьютера распределялся по всему зданию вуза. Это не сервер с Backend системой управления web-контентом, а по сути «плеер» и, несмотря на использование Ethernet кабеля для «доставки» готового видео с этого распределяющего компьютера, доступ к Интернету по этому же каналу (обратно от ТВ) телевизоры не имели. В данном случае для использования встроенного браузера требуется обеспечить дополнительно телевизор Интернетом (или хотя бы доступом к локальному серверу по Ethernet). LAN порт на этих телевизорах свободен, т.к. существует отдельный приёмник сигнала, преобразующий его от распределяющего устройства в обычный порт HDMI. От серверной (Backend) части требуется отдавать контент (например, по user-agent или по параметру url get) в специальном шаблонизаторе, предназначенном для отображения на ТВ. Так как системы Digital Signage не ограничиваются лишь отображением контента на ТВ, всегда можно найти плагины, позволяющие запускать любые браузеры в полноэкранном режиме и заблокировать его контекстное меню, если необходимо задействовать сенсорные информационные киоски. Для пользователя никакого отличия от нативных оболочек интерфейса такого web-приложения иметь не будет.

К настоящему времени отсутствует единый открытый стандарт управления контентом в системах Digital Signage. Существует множество фирм, пытающихся разрабатывать свои программно-аппаратные плееры и облачные SaaS решения, но ни одно из этих решений не позволяет связать базу данных контента имеющегося сайта с той или иной закрытой системой Digital Signage, а решения на основе RSS лент сайтов не являются полноценными.

### Заключение

Представленная в статье методика визуализации контента (и взаимодействия с контентом через сенсорные экраны) посредством HTML, JS, CSS позволяет автоматизировать процесс вывода информации с корпоративного сайта вуза на внутренние ТВ-экраны и «интерактивный киоск», а также избежать избыточного дублирования контента, благодаря использованию только одной базы данных. Здесь реализуется основной принцип работы: «меняется в одном месте – меняется везде», при этом, система управления контентом (CMS) должна поддерживать выгрузку контента в разные структуры (шаблоны) вывода – для сайта и для ТВ.

Как показала практика, информационные табло вуза востребованы среди студентов, преподавателей, сотрудников и посетителей, а решение на основе открытых web-технологий позволяют адаптировать сайт вуза к выводу гипертекста на ТВ с помощью подготовленной структуры вывода. Благодаря набору параметров форматирования CSS один и тот же контент может отображаться одинаково хорошо, как на классических горизонтальных ТВ экранах, так и на вертикальных. Структура вывода при этом может не меняться.

К одному из серьёзных ограничений для предлагаемого в статье технического решения относится зависимость доступа «плееров» в стенах вуза от Интернета и нестабильность хостинга сайта вуза, который должен обрабатывать большие запросы к базе данных. Но это ограничение может быть нивелировано путем развёртывания синхронизируемой базы данных в локальной сети.

Особого внимания заслуживают преимущества описанных в статье открытых стандартов для web, которые предлагаются в качестве отсутствующей альтернативы DS. Такая система может быть свободно модифицирована, без привлечения отдельного специалиста – достаточно знаний разработчика сайта, также нет зависимости от политики компаний-производителей систем Digital Signage. Стандарты web априори означают повсеместную кроссплатформенность и об этом свидетельствует то, что браузеры есть в любом телевизоре, поддерживающем Smart TV. Это позволяет избавиться от аппаратного плеера, подключаемого к ТВ.

Таким образом, несмотря на имеющиеся недостатки, которые были рассмотрены в статье, использование web-технологий позволяет оптимизировать и автоматизировать вывод информационного контента на внутреннем телевидении вуза.

### Список литературы

1. Барсков А. Строим систему Digital Signage // Журнал сетевых решений/LAN [Электронный ресурс]. 2015. URL: <https://www.osp.ru/lan/2015/03/13045266> (дата обращения: 12.05.2018).
2. Барсков А. Тенденции Digital Signage — 2017 // Журнал сетевых решений/LAN [Электронный ресурс]. 2017. URL: <https://www.osp.ru/lan/2017/01-02/13051397/> (дата обращения: 14.05.2018).
3. Компания «Бюро AV решений» [Электронный ресурс]. Тестирование Digital Signage, плееры Brightsign. 201x. URL: [https://av3.ru/article\\_brightsign](https://av3.ru/article_brightsign) (дата обращения: 22.05.2018).
4. Компания «Эколан Тек» [Электронный ресурс]. Приложение мультимедийных сетей HDBaseT. 2015. URL: [http://www.ecolan.ru/imp\\_info/introduction/hdbaset/](http://www.ecolan.ru/imp_info/introduction/hdbaset/) (дата обращения: 25.05.2018).
5. Шевченко Д.А. Конкурентоспособность вуза: методика оценки эффективности сайта вуза в системе Интернет // Вестник РГГУ. Серия «Экономика. Управление. Право». – 2015. – № 3 (146). – С. 111–126.
6. Doyle J. Myth Busting: CSS Animations vs. JavaScript // сайт css-tricks.com [Электронный ресурс]. 2014. URL: <https://css-tricks.com/myth-busting-css-animations-vs-javascript> (дата обращения: 12.05.2018).
7. MDN web docs (Mozilla) [Электронный ресурс]. TouchEvent. 2018. URL: <https://developer.mozilla.org/en-US/docs/Web/API/TouchEvent> (дата обращения: 27.05.2018).
8. Mine Web Design [Электронный ресурс]. URL: <https://minewebdesign.wordpress.com/2013/06/01/workflow-of-php> (дата обращения: 22.05.2018).

### References

1. Barskov A. Stroim sistemu Digital Signage // Zhurnal setevykh reshenij/LAN [Ehlektronnyj resurs]. 2015. URL: <https://www.osp.ru/lan/2015/03/13045266> (data obrashcheniya: 12.05.2018).
2. Barskov A. Tendencii Digital Signage — 2017 // Zhurnal setevykh reshenij/LAN [Ehlektronnyj resurs]. 2017. URL: <https://www.osp.ru/lan/2017/01-02/13051397/> (data obrashcheniya: 14.05.2018).
3. Kompaniya «Byuro AV reshenij» [Ehlektronnyj resurs]. Testirovanie Digital Signage, pleery Brightsign. 201x. URL: [https://av3.ru/article\\_brightsign](https://av3.ru/article_brightsign) (data obrashcheniya: 22.05.2018).
4. Kompaniya «Ehkolan Tek» [Ehlektronnyj resurs]. Prilozhenie mul'timedijnykh setej HDBaseT. 2015. URL: [http://www.ecolan.ru/imp\\_info/introduction/hdbaset/](http://www.ecolan.ru/imp_info/introduction/hdbaset/) (data obrashcheniya: 25.05.2018).
5. SHEVCHENKO D.A. Konkurentosposobnost' vuza: metodika ocenki ehffektivnosti sajta vuza v sisteme Internet // Vestnik RGGU. Seriya «Ehkonomika. Upravlenie. Pravo». – 2015. – № 3 (146). – S. 111–126.
6. Doyle J. Myth Busting: CSS Animations vs. JavaScript // sajt css-tricks.com [Ehlektronnyj resurs]. 2014. URL: <https://css-tricks.com/myth-busting-css-animations-vs-javascript> (data obrashcheniya: 12.05.2018).
7. MDN web docs (Mozilla) [Ehlektronnyj resurs]. TouchEvent. 2018. URL: <https://developer.mozilla.org/en-US/docs/Web/API/TouchEvent> (data obrashcheniya: 27.05.2018).
8. Mine Web Design [Ehlektronnyj resurs]. URL: <https://minewebdesign.wordpress.com/2013/06/01/workflow-of-php> (data obrashcheniya: 22.05.2018).