



ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 51 за 2016 г.



ISSN 2071-2898 (Print)  
ISSN 2071-2901 (Online)

**Бухштаб Ю.А., Воробьев А.А.,  
Евтеева Н.Н.**

Визуальный редактор  
потокowego видео,  
транслируемого с различных  
серверов

**Рекомендуемая форма библиографической ссылки:** Бухштаб Ю.А., Воробьев А.А., Евтеева Н.Н. Визуальный редактор потокowego видео, транслируемого с различных серверов // Препринты ИПМ им. М.В.Келдыша. 2016. № 51. 13 с. doi:[10.20948/prepr-2016-51](https://doi.org/10.20948/prepr-2016-51)  
URL: <http://library.keldysh.ru/preprint.asp?id=2016-51>

**Ордена Ленина  
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ  
имени М.В.Келдыша  
Российской академии наук**

**Ю.А.Бухштаб, А.А.Воробьев, Н.Н.Евтеева**

**Визуальный редактор потокового видео,  
транслируемого с различных серверов**

**Москва — 2016**

*Бухштаб Ю.А., Воробьев А.А., Евтеева Н.Н.*

## **Визуальный редактор потокового видео, транслируемого с различных серверов**

В статье рассматриваются принципы построения программного инструментария, способного поддерживать, на базе использования предлагаемого визуального редактора, процесс создания авторского медиа контента, формируемого из различных распределенных потоковых ресурсов Интернета (прежде всего потокового видео, транслируемого в форматах, используемых в HTML5). Этот контент представляет собой результат виртуального редактирования (объединения различных потоков, удаления фрагментов потоковых данных, вставки в видеопоследовательность новых фрагментов, синхронного наложения на транслируемые видеопоследовательности нового звука или дополнительных элементов и т.д.) и воспроизводится как единое целое или, если это задано, в режиме гипервидео. Описывается организация разработанных программных средств и интерфейс визуального редактора.

**Ключевые слова:** потоковое видео, гипервидео, HTML5

*Yury Alexandrovich Bukhshtab, Andrey Arturovich Vorobiov, Natalia Nikolaevna Evtееva*

## **Visual editor of streaming video broadcasted from different servers**

In the article the principles of creation of software tools capable to support, on the basis of use of the proposed visual editor, the process of creation author's media content formed from various distributed streaming Internet resources are considered (first of all the streaming video broadcasted in the formats used in HTML5). This content is a result of virtual editing (combining different streams, removing the stream data fragments, insertion of new fragments in a video sequence, simultaneous overlay on the video sequence of the new broadcast sound or additional elements, etc.) and is reproduced as a unit or if it is set, in hypervideo mode. The organization of the developed software and the interface of the visual editor are described.

**Key words:** streaming video, hypervideo, HTML5

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект 15-07-00970-а.

В статье рассматривается проблематика разработки программных средств, способных поддерживать полнофункциональное виртуальное редактирование и воспроизведение видео информации, транслируемой с различных распределенных серверов. При этом поддерживается работа в режиме многоуровневого гипервидео. Ранее авторами статьи была разработана структура описания последовательности воспроизводимых фрагментов потоковых данных, позволяющая реализовать поддержку синхронизации медиа потоков и их совместную интерпретацию. Медиа контент, создаваемый из транслируемых распределенных потоков, воспроизводится (в том числе и на мобильных устройствах), как единое целое, разработанным авторами настоящей статьи плеером "Медиа-Гид" (демонстрацию его работы можно посмотреть на странице: <http://www.keldysh.ru/hypervision/demo.html>). В процессе интерактивного воспроизведения результирующего медиа контента плеер поддерживает обработку многих событий, инициирующих вызовы обрабатываемых их функций, результаты выполнения которых становятся доступными в непредсказуемые моменты времени. При функционировании таких приложений эффективная работа может осуществляться только при обеспечении синхронизации между различными процессами. При реализации текущей версии плеера была использована современная технология, стандартизирующая и облегчающая работу с асинхронными процессами. Эта технология базируется на организации асинхронного кода с помощью объектов «обещание» (Promise).

К настоящему моменту в мире разработано несколько программных продуктов, поддерживающих работу с заранее подготовленными потоковыми медиа объектами. В рамках работ по их созданию некоторые из компонент рассматриваемого в статье программного инструментария по отдельности реализованы. Тем не менее программных средств, реализующих набор функциональных возможностей, аналогичный тем, которые описываются в данной статье, в настоящее время не предлагается. Рассмотрим некоторые, наиболее продвинутые, разработки.

Видеохостинг Viddler предоставляет услуги по хранению и трансляции пользовательского видео. Плеер Viddler позволяет воспроизводить склеенные последовательности видео файлов как единый файл. Возможно создание и размещение на экране дополнительных текстовых и графических элементов (не являющихся гиперссылками), воспроизводимых по тайм-коду. Позволяет размещать несколько вариантов видео с различным разрешением и качеством, а также изменять качество во время просмотра с помощью панели управления плеера. Есть система закладок, которые пользователи могут устанавливать во время просмотра. Кроме того, можно создать оглавление видеофайла для быстрого перехода на нужную часть. Создание проекта производится с помощью веб-сервиса Viddler, и там же сохраняется созданный проект. Первоначально был ориентирован только на FLASH. Недавно появилась версия для видео в HTML5.

H5P представляет собой плагин (набор библиотек на JavaScript) для некоторых систем управления контентом (CMS), таких как WordPress, Drupal, Joomla. Предполагается, что видеофайлы хранятся на серверах и доступом к ним управляют сайты, созданные с помощью соответствующей CMS. На серверах этих сайтов сохраняются и созданные проекты. Используются только возможности видео HTML5. Есть возможность использовать альтернативные видео файлы для разных браузеров (в форматах Mp4 или Webm). Плеер H5P позволяет выводить на экран текстовые, графические элементы, а также формы с вопросами. Элементы выводятся на экран в виде небольших иконок и раскрываются при клике на них. Формы с вопросами не предполагают какого-либо взаимодействия с системой, кроме сообщений о том, верный или неверный ответ получен. Возможно создание оглавлений видео файла с помощью заранее подготовленных закладок. Текстовые и графические элементы не являются гиперссылками на другие Web страницы. Все элементы имеют стандартный вид и не предполагают возможности изменения стилового оформления.

Наиболее развитая разработка Popcorn (сообщество Mozilla), представляет собой интерактивную среду для создания проектов интерактивного видео. Может использовать видеофайлы, транслируемые с различных серверов, включая популярные видеохостинги Youtube и Vimeo. Позволяет создавать на экране плеера области с дополнительной текстовой и графической информацией. Эта информация может являться гиперссылкой на другую Web страницу, которая открывается в этом или дополнительном окне браузера. Текстовые и графические элементы могут в некоторых пределах иметь различное стиловое оформление (цвет, шрифт, прозрачность, эффекты появления и исчезновения метки). Видео файл может быть подвергнут редактированию. Можно задать вырезанные (не показываемые) участки, а также зацикленные (повторяемые заданное число раз). Созданные проекты сохраняются на сервере Popcorn в виде HTML страниц и могут быть вставлены в произвольную страницу пользователя в виде фреймов. Кроме того, существует API (в виде библиотек на JavaScript), позволяющий создавать гипервидео файлы вручную, без использования интерактивной среды, непосредственно на HTML странице. Однако использование API требует большого объема программирования на HTML и JavaScript, так как для каждого события (появления элемента на экране) требуется написать функцию на JavaScript, которая будет вызвана в соответствующий тайм-коду момент времени, и сам элемент задать в виде HTML кода.

Отметим теперь некоторые важные преимущества предлагаемого авторами статьи программного инструментария по сравнению с уже существующими в мире продуктами.

Возможно воспроизведение не только отдельных видео файлов, но и последовательности таких файлов, а также заданных тайм-кодами их фрагментов (в том числе вставленных в другой видео файл) как единой

видеопоследовательности. Кроме этого, возможно воспроизведение графических изображений в режиме слайд-шоу. Также возможно параллельное и синхронное воспроизведение видео и аудио потоков, из различных файлов и с различных серверов. Можно использовать видео файлы, размещенные на любых HTTP серверах. Дополнительные элементы медиа контента могут содержать произвольную информацию – текстовую, графическую или в виде HTML форм (с полями ввода, выбора, кнопками). При этом для оформления элементов могут быть использованы все возможности HTML и CSS.

Активность элемента может быть вызвана не только по клику мыши, но и другими событиями. В качестве реакции на действия пользователя будет возможно не только открытие другой Web страницы, но и показ дополнительной информации на экране плеера, изменение информации на текущей странице. Кроме этого, имеется возможность перейти на другой тайм-код в текущем клипе или начать показ другого клипа, с возможностью вернуться назад. В качестве реакции элементу можно назначить несколько действий.

Наконец, очень важно то, что предлагаемая авторами настоящей статьи методология основана на новом подходе, который предполагает объединение средств, обеспечивающих создание единого медиа контента, представляющего собой результат виртуального склеивания, удаления, выделения и переупорядочивания видео и аудио файлов и их фрагментов со средствами задания и воспроизведения целевого контента в режиме гипервидео.

Создаваемый авторский медиа контент (будем называть его проектом) передается плееру в виде файла, задающего управление воспроизведением этого контента. Этот файл содержит информацию о воспроизводимых видеопоследовательностях (клипах), их фрагментах, аудиопотоках, графических изображениях и дополнительных элементах. Эта информация представлена в файле в формате XML или JSON и может размещаться на любых серверах, независимо от того, где расположены видео файлы.. Такой файл может быть создан вручную в текстовом редакторе. Однако это требует определенной квалификации и знания не самой простой структуры описания элементов проекта. Гораздо проще создать файл проекта, используя интерактивную программу, позволяющую задавать структуру проекта из клипов и фрагментов нажатием кнопок, перетаскиванием элементов мышью, и задавая необходимые значения в строго определенных полях. Поэтому встал вопрос о разработке такой программы (визуального редактора), использование которой позволяет пользователю задавать последовательности воспроизведения медиа контента и структуры гипервидео возможно не только вручную, но и путем создания и редактирования XML или JSON файла с помощью развитой интерактивной среды.

Реализующая редактор программа является Web приложением, исполняемым в среде любого браузера, поддерживающего стандарт HTML5. Она представлена набором файлов, содержащих описание страниц браузера в

формате HTML, описаний стилей в формате CSS и программ на JavaScript. Эти файлы могут быть размещены на удаленном HTTP сервере или на локальном компьютере пользователя.

В терминологии приложений в сетях с клиент-серверной архитектурой программу создания гипермедиа проектов можно назвать «толстым клиентом». Соответствующий англоязычный термин – «rich client». Так называют приложения, где основная функциональность сосредоточена на компьютере пользователя, а сервер может использоваться как хранилище для программ и, может быть, некоторых данных. С сервера клиенту передаются только программы на JavaScript и HTML страницы без наполнения данными, фактически заготовки. Во время исполнения JavaScript программ браузер может получить необходимые данные с сервера или из другого источника. Обычно такие данные передаются в виде XML или JSON файлов. Программа производит обработку этих данных и формирует их представление в окне браузера, полностью строя необходимую HTML страницу, при этом могут использоваться заготовки. Подобные «толстые клиенты» могут иметь большую функциональность, чем обычные Web страницы.

Во время работы программе создания гипермедиа проектов может вообще не требоваться получать или сохранять на сервере какие-либо данные. Вся работа производится в среде браузера на компьютере пользователя. Все результаты – конечные и промежуточные – сохраняются на компьютере пользователя. Для этого используются средства, имеющиеся в современных браузерах. Одним из таких средств является локальное хранилище (local storage). Локальное хранилище позволяет программе на JavaScript сохранять довольно большой объем данных на компьютере пользователя между сессиями. Сохраненная информация доступна только страницам, загружаемым с одного домена, а в случае если страница загружается с локального компьютера, информация доступна страницам из одного каталога. Информация в локальном хранилище хранится в виде пар «ключ» «значение», где ключ и значение – произвольные текстовые строки. Несмотря на такое ограничение, программа способна сохранять в локальном хранилище практически любую структурированную информацию, представляя ее, например, в виде строки в формате JSON.

При работе с визуальным редактором в локальном хранилище хранится создаваемый медиа проект. Все изменения, вносимые пользователем, автоматически сохраняются в локальном хранилище. Если пользователь закончит работать с проектом, то когда он снова в следующий раз запустит программу, все результаты предыдущей работы будут находиться в хранилище. Пользователю также предоставляется возможность записать создаваемый проект в файл на своем компьютере из хранилища или прочитать ранее сохраненный файл проекта. Такие возможности предоставляет стандарт HTML5. Подобные операции браузером не могут производиться автоматически, так как считаются небезопасными. Они требуют

подтверждений от пользователя на совершение таких операций. Используя эти возможности, пользователь может сохранять промежуточные результаты, например, чтобы продолжить работу на другом компьютере или в другом браузере. Медиа проект может быть сохранен или считан как в формате JSON, так и XML.

При создании проекта для медиа плеера, обладающего дополнительными возможностями, предоставляемыми технологией гипервидео, с помощью программы – визуального редактора, пользователь оперирует объектами, аналогичными тем, которые описываются в файле проекта при создании его вручную: клипами, потоками, фрагментами, дополнительными элементами. Для каждого объекта в программе имеется соответствующее окно, содержащее информацию об этом объекте. Для таких объектов, как клип, поток, фрагмент в этом же окне содержится экран для просмотра объекта, независимо от других. Информация об объекте содержится в текстовых полях окна, в которые пользователь может вводить и редактировать информацию. Кроме того, в окнах имеется ряд кнопок, с помощью которых пользователь может выполнять над объектом такие операции, как создание, сохранение, удаление. Для выполнения многих операций может быть использована технология Drag&Drop. Так, для воспроизведения клипа или фрагмента в окне просмотра можно просто перетянуть мышью имя нужного клипа из списка клипов или имя нужного фрагмента из списка фрагментов на экран просмотра. А для задания потока можно перетянуть имя потока из списка потоков в нужное поле.

Для объектов, которые могут быть воспроизведены, в окне имеется экран предварительного просмотра. Этот экран является экраном медиа плеера для воспроизведения только указанного объекта. При работе с окнами объектов пользователь имеет возможность перемещать окна по экрану в пределах окна браузера, чтобы можно было расположить их в удобном порядке, закрывать ненужные. Некоторые окна могут быть расширены или сужены, чтобы раскрыть или скрыть дополнительную информацию или использовать дополнительные возможности. Основное окно проекта содержит список клипов, входящих в проект. Первый клип в списке является основным. С него начинается воспроизведение в плеере. Остальные клипы – вспомогательные и могут быть воспроизведены при переходе на них из дополнительных элементов.

Клипы в списке отображаются своими именами. Пользователь может выбирать клип и выполнять следующие операции, используя расположенные в окне кнопки:

- открыть клип для редактирования;
- воспроизвести клип на экране предварительного просмотра;
- удалить клип.

Имеется также кнопка создания нового клипа.

При создании нового клипа или редактировании существующего открывается окно клипа. В этом окне отображается информация о выбранном

клипе: имя клипа, список потоков, используемых в клипе, список фрагментов, список дополнительных элементов, относящихся к клипу. Используя расположенные в окне кнопки, пользователь может создавать новые, редактировать и удалять элементы списков. При этом открывается новое окно потока, фрагмента или дополнительного элемента.

Потоки и фрагменты в списках отображаются своими именами, их можно просматривать в окне предварительного просмотра. Фрагменты в списке расположены в том порядке, в котором они воспроизводятся в клипе. Пользователь имеет возможность, используя специальные кнопки, перемещать выбранный фрагмент вверх или вниз по списку или просто переставить фрагмент в нужное место в списке, используя технологию Drag&Drop.

Дополнительные элементы должны воспроизводиться в точно заданные моменты времени. Их начальный тайм-код всегда отсчитывается от начала клипа и не связан с входящими в клип фрагментами. Дополнительные элементы представлены в соответствующем списке своими именами и тайм-кодами. Они расположены в этом списке в порядке увеличения тайм-кодов.

Окно потока открывается, когда пользователь создает новый или редактирует существующий поток. При создании потока необходимо задать имя потока, которое используется для ссылок на него. Имя задается только при создании потока и не может быть изменено в дальнейшем. Если имя не задано, программа автоматически назначит потоку уникальное имя. В окне потока содержится список URL – адресов видео или аудио файлов, представляющих поток. В потоке могут быть заданы файлы, содержащие одинаковую аудио или видео информацию, но закодированную в различных форматах (Mp4, Webm, Ogg, Mp3). Это связано с тем, что различные браузеры поддерживают различные форматы. При воспроизведении потока браузер сам может выбрать из списка файл того формата, который он может воспроизвести. Пользователь может добавлять новые, редактировать или удалять существующие файлы в списке. Кроме того, в качестве потока может быть задана ссылка на ресурс видеосервиса YouTube.

Окно фрагмента открывается, когда пользователь создает или редактирует фрагмент в клипе. При создании фрагмента необходимо задать имя фрагмента и выбрать тип фрагмента: видеофрагмент или галерея. Имя фрагмента также задается только при создании и служит для ссылок на него. Его нельзя изменить впоследствии. Если имя не задано, программа автоматически назначит фрагменту уникальное имя.

В зависимости от выбранного типа информация, представляемая в окне фрагмента, будет различной. Так, для видеофрагмента пользователь должен назначить один из ранее созданных потоков. Для этого можно ввести имя в соответствующее поле или, используя операцию Drag&Drop, перетащить в это поле нужный поток из списка потоков в окне клипа.

Видеофрагмент содержит начальный и конечный тайм-коды фрагмента, выбранного из потока. Установить эти значения можно вручную, введя тайм-

коды в соответствующие поля. Но можно, используя экран окна предварительного просмотра, остановить просмотр потока в нужном месте и отметить это место как начальный или конечный тайм-коды с помощью соответствующих кнопок. Имеются также кнопки точного позиционирования, которые позволяют, находясь в режиме паузы, передвигать текущий тайм-код вперед или назад на небольшие промежутки (например, 1 или 0,2 секунды).

Фрагмент типа галерея содержит список графических изображений, из которых состоит фрагмент. Изображения в списке представлены своими миниатюрами – уменьшенными изображениями, URL адресами и временем воспроизведения. Пользователь может добавлять новые изображения в список, редактировать и удалять существующие. Для этого необходимо ввести URL адрес изображения в список и задать время воспроизведения. Возможно использование операции Drag&Drop для ввода URL адреса, надо просто перетащить нужное изображение из другого окна браузера в поле URL адреса.

При воспроизведении фрагмента изображения показываются в том порядке, в котором они заданы в списке. Пользователь может менять порядок изображений, используя специальные кнопки и перемещая выбранное изображение вверх или вниз по списку, или, используя операцию Drag&Drop, просто переместить выбранное изображение в нужное место в списке.

На фрагмент может быть наложен дополнительный аудио поток. Для этого необходимо в соответствующее поле ввести имя потока или перетащить, используя Drag&Drop, нужный поток из списка. В поле тайм-кода аудио потока можно задать тайм-код, с которого начнется воспроизведение звукового потока при воспроизведении данного фрагмента. Также возможно установить в соответствующих полях относительные уровни громкости аудио и видео частей фрагмента.

Окно фрагмента содержит список дополнительных элементов, относящихся к фрагменту. Тайм-коды начала этих элементов задаются относительно начала потока, из которого выбран фрагмент, или относительно начала фрагмента. Пользователь может создавать новые, редактировать и удалять существующие элементы, используя расположенные в окне кнопки. При создании и редактировании дополнительного элемента открывается новое окно дополнительного элемента.

Окно дополнительного элемента открывается, когда пользователь создает или редактирует дополнительный элемент, связанный с клипом или фрагментом. В окне элемента имеется множество полей, описывающих элемент:

- имя элемента;
- тип элемента – графический или текстовый;
- координаты элемента в окне плеера;
- текст элемента;
- URL графического изображения;
- тайм-код начала;

- продолжительность воспроизведения;
- строка стилевого оформления элемента;
- флажок паузы.

Имя дополнительного элемента необязательно и предназначено для ссылок на него. Оно задается при создании элемента и не может быть изменено впоследствии. Ссылка на дополнительный элемент требуется, если данный элемент необходимо воспроизводить или убирать с экрана в качестве реакции на какие-либо действия пользователя.

Координаты, определяющие положение дополнительного элемента на экране, могут быть заданы числовыми значениями в соответствующих полях, или пользователь может мышью выделить на экране плеера в окне предварительного просмотра прямоугольную область, в которой будет расположен дополнительный элемент. Эта область будет отмечена особой границей на экране плеера и впоследствии, при редактировании данного элемента, ее можно изменять с помощью мыши. При числовом задании координат нужно использовать такие параметры, как координаты краев области, ее ширину и высоту. Необходимо указывать две величины для горизонтального и вертикального размещения. Например, для горизонтального размещения можно задать левую координату и ширину, левую и правую координату, правую координату и ширину. Следует отметить, что левая координата задается расстоянием от левой границы видеоизображения на экране до левой границы дополнительного элемента, а правая координата задается расстоянием от правой границы видеоизображения на экране до правой границы дополнительного элемента. Аналогично, координата верхней границы дополнительного элемента задается как расстояние от верхнего края видеоизображения до верхнего края дополнительного элемента, а нижняя координата как расстояние от нижнего края видеоизображения до нижнего края элемента.

Для элементов, относящихся к клипу, тайм-код начала воспроизведения элемента задается относительно начала всего клипа. Для элемента, относящегося к фрагменту, тайм-код начала может быть задан как относительно начала потока фрагмента, так и относительно начала самого фрагмента. Имеется специальный флажок, указывающий, откуда отсчитывать тайм-код начала.

Существует три способа указать время, когда на экране появится дополнительный элемент. При этом этот элемент может относиться как к клипу, так и к фрагменту.

Если дополнительный элемент описывается внутри клипа, но не внутри фрагмента (на XML он вложен внутри тегов `<clip>... </clip>`), то он относится к клипу и его тайм-код отсчитывается от начала клипа, независимо от того, какие там фрагменты.

Если дополнительный элемент описывается внутри фрагмента (на XML он вложен внутри тегов `<fragm>... </fragm>`), то он относится к фрагменту. В этом

случае время его появления на экране должно отсчитываться от начала воспроизведения фрагмента. Но так как начало фрагмента задается относительно начала видео файла, то можно задать появление дополнительного элемента как относительно начала видео файла (параметр `begin` в XML), так и относительно начала фрагмента в клипе (параметр `fragmBegin` в XML). Это зависит от того, как в каждом конкретном случае удобнее. Например, если пользователь хочет поставить ссылку для перехода на сайт в Интернете, в котором содержится описание объекта появляющегося в текущем фрагменте, то ему удобнее задать начало дополнительного элемента относительно начала видео файла, из которого берется фрагмент. Как бы потом ни двигали начало фрагмента, тайм-код появления элемента меняться не будет. А если мы хотим спустя 3 секунды после начала фрагмента дать возможность пропустить его, то удобнее задать тайм-код дополнительного элемента относительно начала фрагмента.

Также тайм-код начала можно задать числом в соответствующем поле или, используя экран предварительного просмотра, остановить воспроизведение фрагмента в нужном месте и специальной кнопкой задать начальный тайм-код. Аналогично необходимо задать конечный тайм-код или продолжительность показа дополнительного элемента.

В строке стилевого оформления можно задать некоторые параметры, определяющие внешний вид дополнительного элемента. Строка задается в виде, принятом для задания стилевого оформления в HTML и CSS:

```
<параметр> : <значения>; <параметр> : <значение>;...
```

Можно использовать различные параметры, принятые в CSS. Так, параметр `color` задает цвет текста, `background` – цвет фона, `border` – цвет и ширину границы, а параметром `font` можно задать тип и размер шрифта.

Флажок паузы показывает, нужно ли переводить плеер в режим паузы при появлении на экране данного дополнительного элемента.

В окне дополнительного элемента расположен список, задающий реакцию на различные события, связанные с действием пользователя над дополнительным элементом. Элемент списка содержит событие, заданное своим именем, и действия, которые должны быть выполнены при наступлении этого события.

Имя события выбирается из списка и может быть одним из следующих:

- Click – щелчок по элементу,
- Enter – приход курсора мыши в элемент,
- Leave – выход курсора мыши из элемента,
- TimeCode – достижение тайм-кода.

События Click, Enter и Leave наступают, когда пользователь совершает соответствующие действия над элементом. А событие TimeCode возникает, когда дополнительный элемент появляется на экране, то есть плеер достигает тайм-кода, заданного как начальный для данного элемента.

Каждый элемент списка событий содержит множество полей, в которых задаются реакции на наступление данного события. Имеются следующие поля:

- поля перехода на другой тайм-код;
- поле цели;
- поле изменяемого URL;
- поле изменяемого текста;
- поле изменяемого стиля;
- поле высвечиваемых элементов;
- поле гасимых элементов.

Переход на другой тайм-код задается с помощью полей, задающих клип, на который необходимо перейти, фрагмент в клипе и тайм-код. Клип можно выбрать из списка имен существующих клипов. Если не задать клип, то переход будет осуществляться в пределах текущего клипа. Фрагмент выбирается из списка имен фрагментов заданного или текущего клипа. Тайм-код задает переход на задаваемую им позицию, считая от начала фрагмента, если он задан, или начала клипа, если фрагмент не задан. Кроме того, можно использовать флажок паузы, который указывает, что после перехода плеер должен перейти в состоянии паузы до нажатия пользователем кнопки PLAY.

Кроме перехода на другой тайм-код, при наступлении события могут быть изменены другие дополнительные элементы. Поле цели задает имя дополнительного элемента, который должен быть изменен. Имя выбирается из списка имен заданных дополнительных элементов. Поле изменяемого URL задает адрес нового изображения, которое будет высвечено для заданного элемента. Поле изменяемого текста задает новый текст, который будет высвечен для элемента. Поле стиля задает новый стиль для элемента.

Если не задано поле цели, но задан новый URL, то браузер откроет новое окно (новую вкладку) с заданным URL адресом.

В качестве реакции на события можно высветить или убрать с экрана другие дополнительные элементы. Эти элементы задаются в полях высвечиваемых и гасимых дополнительных элементов. В каждом поле задается строка имен элементов, разделенных пробелом.

К дополнительному элементу может быть привязано несколько элементов с одинаковым именем события. Например, если требуется изменить стиль нескольких дополнительных элементов при щелчке мыши на данном, то для каждого из них необходимо задать реакцию на такое действие.

В результате использования визуальный редактор формирует файл, задающий управление воспроизведением (в том числе и в режиме гипервидео) авторского медиа контента плеером "Медиа-Гид".

### Литература

1. Frank Shipman, Andreas Girgensohn, Lynn Wilcox. Hypervideo Expression: Experiences with Hyper-Hitchcock // Sixteenth ACM Conference on Hypertext and Hypermedia. 2005. URL: <http://www.fxpal.com/publications/hypervideo-expression-experiences-with-hyper-hitchcock.pdf>
2. Anna Weston. The Evolution of Hypervideo // Electronics and Computer Science University of Southampton. 2013. URL: [http://mms.ecs.soton.ac.uk/2013/papers/aw3g10\\_23531509\\_finalpaper.pdf](http://mms.ecs.soton.ac.uk/2013/papers/aw3g10_23531509_finalpaper.pdf)
3. HTML5 – A vocabulary and associated APIs for HTML and XHTML. – W3C Recommendation 28 October 2014. – URL: <https://www.w3.org/TR/2014/REC-html5-20141028/>
4. Бухштаб Ю.А., Воробьев А.А., Евтеева Н.Н. Реализация программных средств, обеспечивающих управление доставкой видео и аудио данных на базе HTML5 и Flash // Препринты ИПМ им. М.В.Келдыша. 2012. № 62. URL: <http://library.keldysh.ru/preprint.asp?id=2012-62>
5. Бухштаб Ю.А., Воробьев А.А., Евтеева Н.Н. Интерактивные возможности управления потоковым видео в среде HTML5 // Препринты ИПМ им. М.В.Келдыша. 2013. № 36. URL: <http://library.keldysh.ru/preprint.asp?id=2013-36>