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

Препринты ИПМ • Препринт № 36 за 2013 г.

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

Интерактивные возможности управления потоковым видео в среде HTML5

Рекомендуемая форма библиографической ссылки: 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. 19 с. URL: http://library.keldysh.ru/preprint.asp?id=2013-36

О р д е н а Л е н и н а
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Р о с с и й с к о й а к а д е м и и н а у к

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

Интерактивные возможности управления потоковым видео в среде HTML5

Москва — 2013

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

Интерактивные возможности управления потоковым видео в среде HTML5

В работе рассматривается набор управляющих структур на языке XML, предназначенных для обеспечения интерактивного взаимодействия пользователя с плеером в процессе воспроизведения потоковых видеоданных. Эти структуры интерпретируются программными компонентами, реализованными в среде HTML5, поддерживая возможность использования технологии гипервидео для управления потоковыми мультимедийными данными, транслируемыми с распределенных серверов и представленными в различных форматах.

Ключевые слова: мультимедиа, потоковое видео, HTML5, гипервидео

Yury Alexandrovich Bukhshtab, Andrey Arturovich Vorobiov, Natalia Nikolaevna Evteeva

Interactive possibilities of controlling streaming video in the environment of HTML5

This paper describes the set of control structures on XML which is purposed to provide an interactive user work with the player during playback of streaming video. These structures are interpreted by software components implemented in the environment of HTML5, supporting the use of hypervideo technology for managing streaming multimedia data services transmitted from distributed servers and presented in various formats.

Key words: multimedia, streaming video, HTML5, hypervideo

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект 11-07-00258-а.

В статьях [1,2] рассматривалась реализация программных средств, поддерживающих предложенную авторами настоящей работы информационную технологию управления видео- и аудиопотоками, транслируемыми различными территориально разнесенными серверами. Используя такие программные средства, пользователи могут создавать свои собственные потоковые видеопоследовательности путем объединения различных потоков и их виртуального редактирования, например, выделения фрагментов потоковых данных, имплантации в видеопоследовательность новых фрагментов, удаления частей потока, синхронного наложения на транслируемые видеопоследовательности нового звука и т.д. Полученный в результате таких действий видео- и/или аудиопоток воспроизводится клиентской программой, которую далее будем называть плеером. Разработанный плеер способен функционировать в средах всех основных браузеров, воспроизводя как единое целое различные видео- и аудиопотоки и их фрагменты, в частности, закодированные в форматах, предназначенных для воспроизведения медиаэлементами браузеров, поддерживающих HTML5 (MPEG, WEBM, MP3, OGG).

На практике часто бывает недостаточно только воспроизводить видео- и аудиоинформацию. Так, иногда необходимо дать какую-либо пояснительную текстовую информацию или дополнительную графическую информацию. Такие средства входят в состав разработанного программного обеспечения. Они позволяют в заданные моменты времени выводить в определенных местах экрана дополнительные текстовые или графические элементы. Например, это позволяет воспроизводить на экране субтитры или в фильме, показывающем городскую инфраструктуру, в нужный момент времени вывести на экране текст, поясняющий, какой именно объект сейчас демонстрируется, или разместить в углу экрана дополнительное фотоизображение этого объекта в другом ракурсе.

Однако во многих предметных областях при использовании возможностей мультимедиа и таких средств недостаточно. Так, например, в образовательных приложениях часто требуется дать возможность пользователю изменять последовательность просмотра видео, получить дополнительную информацию, не прерывая просмотра, пропустить часть необязательной информации и т.д. То есть возникает необходимость обеспечить интерактивное взаимодействие пользователя с плеером в процессе воспроизведения потоковых видеоданных. Другими словами, речь идет о реализации технологии гипервидео, которая основана на понятии гиперссылки, учитывающем временные и пространственные аспекты видео. Ранее созданные авторами программные средства виртуального управления видео- и аудиопотоками обеспечивают поддержку синхронизации потоковых данных в среде HTML5, аналогичную по своим возможностям имеющейся в языке SMIL (SMIL не обеспечивает этой поддержки в среде HTML5). Такие возможности предоставляют необходимый

инструмент для полнофункционального использования альтернативного контента.

Плеер позволяет связать с дополнительными текстовыми и графическими элементами определенную активность, а именно, назначить дополнительные действия, которые будут выполнять программы при наступлении каких-то событий. Событиями могут быть щелчок мышью на дополнительном элементе, перемещение курсора мыши в область или из области элемента. Достижение определенного тайм-кода также можно объявить событием, хотя это событие и не связано с действием пользователя.

При наступлении события плеер может выполнять следующие действия:

— перейти на заданный тайм-код внутри воспроизводимой видеопоследовательности;

— начать воспроизводить другую видеопоследовательность с возможностью вернуться в прерванное место исходной;

— показать на экране плеера дополнительную текстовую или графическую информацию;

— открыть дополнительное окно браузера с заданным URL-адресом.

Описанные выше действия никак не связаны с HTML страницей, в которую встроен плеер, и могут выполняться на любой странице. Если предполагается использование плеера в рамках какого-либо конкретного проекта и специальным образом подготовлены страницы, на которых он будет размещен, то можно использовать некоторые дополнительные действия, выполняемые плеером. В этом случае при наступлении событий плеер может:

— поменять изображение в заранее определенном элементе <IMG> на странице;

— поменять содержимое заранее определенного элемента <IFRAME> по заданному URL-адресу;

— в произвольный заранее определенный элемент на HTML странице поместить информацию из другого элемента (например, заранее созданного невидимого элемента <DIV>);

— поместить информацию из произвольно элемента HTML страницы (например, заранее созданного невидимого элемента <DIV>) на экран плеера;

— выполнить заранее определенную на HTML странице функцию JavaScript.

Такие возможности могут иметь много применений. Например, при воспроизведении видеофильма о городских достопримечательностях можно на странице выводить карту города с указанием этих объектов, по запросу пользователя показывать дополнительные фотоизображения этих объектов. В учебных фильмах можно дать возможность пропускать излишне подробную информацию или, наоборот, по желанию пользователя дать дополнительную информацию. Можно контролировать усвоение и понимание материала,

задавая вопросы, и в случае неверного ответа предлагать дополнительный материал.

Таким образом, плеер может работать с достаточно обширной и разнородной информацией. Для описания такой информации авторами статьи были разработаны специализированные структуры на языке XML с наборами необходимых тегов и атрибутов. Эти структуры интерпретируются входящими в состав плеера программными компонентами. Плееру передается URL адрес XML файла, в котором определяется последовательность воспроизведения фрагментов видео- и аудиофайлов, содержатся описания дополнительных видеопоследовательностей, воспроизводимых по запросу пользователя, дополнительных текстовых и графических элементов и определяется их активность.

XML файл должен начинаться с пролога, указывающего тип файла и кодировку

```
<?xml version="1.0" encoding="UTF-8"?>
```

Файл должен содержать корневой тег <play>, внутри которого расположены другие теги, описывающие информацию на воспроизведение.

```
<play>
    ... другие теги
</play>
```

тег <layout /> определяет размер плеера на странице

```
<layout  width="число" height="число" />
```

Атрибуты width и height задают ширину и высоту экрана плеера в пикселях соответственно.

Пример:

```
<layout  width="640" height="480" />
```

Для воспроизведения видео- и аудиоинформации браузерами, поддерживающими HTML5, могут использоваться различные типы кодировок файлов. К сожалению, браузеры не способны воспроизводить все виды форматов кодировок. В некоторых случаях у разных браузерах имеется полное несовпадение воспроизводимых форматов. Поэтому для трансляции видео и аудио в HTML5 обычно осуществляется кодирование одного фильма в несколько форматов, и видеоэлементу передаются имена нескольких файлов, содержащих одну и ту же информацию в различных форматах кодировки, а браузер сам выбирает какой именно формат он может воспроизвести.

Такая возможность реализуется с помощью тега <stream>, внутри которого тегами <file> описываются файлы различных форматов для одного и того же видео.

Тег <stream> имеет вид

<stream name="строка"> </sream>

Единственный атрибут name задает имя тега для последующих ссылок на него. Имя должно быть уникальным, состоять только из латинских букв и цифр и начинаться с буквы. Внутри тега <stream> могут быть расположены один или несколько тегов <file>, которые описывают различные имеющиеся форматы кодировки.

Тег имет вид

<file src="url адрес" type="mime тип" />

Обязательный атрибут src задает адрес размещения файла. Атрибут type определяет вид кодирования информации в этом файле, указывая его MIME тип, например "video/mp4", "audio/ogg" и т.д. Обычно плеер сам может определить тип кодировки по расширению файла, и в этом случае указывать атрибут type необязательно. Но иногда бывают ситуации, когда используются нестандартные расширения файлов, например mpeg для файлов, закодированных в формате mp4, или используется динамическая генерация файлов серверными программами. В таких случаях атрибут type необходимо задавать.

Пример:

```
<stream name="film1">
    <file src="www.myserv.net/media/film-01.webm" />
    <file src="www.myserv.net/ media/film-01.mpg" type="video/mp4" />
</stream>
```

Непрерывная последовательность, воспроизводимая плеером, называется клипом. В XML файле должен быть описан один или несколько клипов. Воспроизведение начинается с первого клипа, остальные клипы могут быть воспроизведены в качестве дополнительной информации при активизации дополнительного элемента. Клипы описываются тегами <clip>, которые располагаются внутри корневого тега <play>. Тег имеет вид:

<clip name="строка"> </clip>

Атрибут name определят имя клипа и используется для ссылок на него. Имя должно быть уникальным, состоять из латинских букв и цифр и начинаться с буквы.

Таким образом описывается последовательность видеофрагментов, которые должны быть воспроизведены непрерывно. Видеофрагментом может быть фрагмент видеофайла, заданный своим адресом и тайм-кодами начала и конца, или галерея, состоящая из одного или нескольких фотоизображений, которые воспроизводятся в режиме слайд-шоу. На видеофрагменты может быть наложен дополнительный аудиофрагмент, который воспроизводится параллельно и синхронно с видеофрагментом. Аудиофрагмент также задается адресом файла и тайм-кодами начала и конца. Кроме видеофрагментов внутри тега <clip> могут быть размещены теги, описывающие дополнительные элементы, относящиеся к клипу.

Видео- и аудиофрагменты описываются тегами <video> и <audio> соответственно. Эти теги имеют следующие атрибуты:

name — задает имя фрагмента для ссылок на него;

stream — указывает имя одного из потоков, определенных тегами <stream>, для указания файла, который будет воспроизводиться;

src — указывает адрес файла, который будет воспроизводиться, если нужно воспроизвести именно этот файл, без выбора с помощью <stream>;

type — определяет MIME-тип файла, заданного атрибутом src; при описании необходимо указывать либо атрибут stream, либо атрибуты src и type;

volume — указывает начальный уровень громкости при воспроизведении фрагмента, задается целым числом от 0 (без звука) до 100 (максимальная громкость); если не указан, то используется текущий установленный в плеере уровень;

begin — задает начальный тайм-код фрагмента;

end — задает конечный тайм-код фрагмента;

dur — задает продолжительность фрагмента.

Не обязательно указывать все атрибуты begin, end и dur. Достаточно указать те, которые однозначно определяют начало и конец фрагмента, например, только begin и end, или begin и dur, или end и dur. Если атрибуты не указаны или указаны только атрибуты end или dur, то файл воспроизводится с

начала. Если атрибуты не указаны или указан только атрибут begin, то файл воспроизводится до конца.

Тайм-коды и продолжительность можно указывать в различных форматах:

— в виде вещественного числа, задающего число секунд;

— в виде целого числа с символами "ms" в конце, задающего число миллисекунд;

— в виде строки вида "hh:mm:ss.dd" ,где hh — часы, mm — минуты ss — секунды, dd — десятые доли секунды. Если часы или десятые доли секунды равны нулю, то их можно не указывать.

Так, записи begin="128.7", begin="128700ms" и begin="2:08.7" определяют один и тот же тайм-код.

Кроме того, внутри тегов <video> и <audio> располагаются теги <file>, аналогично как при использовании тега <stream>. При этом плеер выберет для воспроизведения один из указанных файлов. Атрибут stream при этом не указывается.

Примеры:

```
<video name="frag1" stream="film1" begin="5:21" dur="90" volume="60" />

<video begin="15:36">
    <file src="www.myserv.net/ media/film-01.webm" />
    <file src="www.myserv.net/ media/film-01.mpg" type="video/mp4" />
</video>
```

Видеофрагмент типа «галерея» описывается тегом <gallery>, внутри которого расположены теги <image>, описывающие изображения. Тег <gallery> может иметь атрибут name для указания имени фрагмента. Тег <image> описывает графический файл и имеет атрибуты:

src — адрес файла фотоизображения,

dur — время, в течение которого изображение воспроизводится на экране.

Пример:

```
<gallery name="fragm5">
      <image src="/photo/fragm5/ph-1.jpg" dur="5"/>
      <image src="/photo/fragm5/ph-2.jpg" dur="8"/>
      <image src="/photo/fragm5/ph-3.jpg" dur="5"/>
</gallery>
```

Чтобы задать наложение аудиофрагмента на видеофрагмент, используется тег <par>, задающий параллельное воспроизведение фрагментов. Тег имеет атрибут name, определяющий имя фрагмента. Внутри тега <par>

</par> размещаются теги видео- и аудиофрагментов, которые надо воспроизвести параллельно.

Пример:

```
<par name="frag8">
    <video begin="75.4" end="5:13" volume="20"/>
        <file src="www.myserv.net/ media/sea.webm" />
        <file src="www.myserv.net/ media/sea.mpg" type="video/mp4" />
    </video>
    <audio stream="seasound" begin="48" volume="80" />
</par>
```

Продолжительность всего параллельного фрагмента определяется по продолжительности видеофрагмента. Атрибуты volume в параллельном фрагменте имеют немного отличное значение от обычного видеофрагмента. Начальный уровень громкости определяется атрибутом volume у аудиофрагмента. А атрибут volume у видеофрагмента устанавливает его уровень громкости в процентах от текущего уровня аудиофрагмента.

Для воспроизведения на экране плеера дополнительных элементов используется тег <extelm>. Этот тег может быть помещен внутри тега <clip></clip> или внутри тегов, описывающих фрагменты (<video>, <gallery> или <par>). Тег <extelm> имеет много атрибутов, определяющих тип, вид и место размещения дополнительного элемента.

Атрибут name определяет имя дополнительного элемента для ссылок на него.

Имеется набор атрибутов, определяющих место расположения элемента на экране плеера:

width — задает ширину элемента;

height — задает высоту элемента;

left — задает расстояние левой границы элемента от левого края экрана плеера;

right — задает расстояние правой границы элемента от правого края экрана плеера;

top — задает расстояние верхней границы элемента от верхнего края экрана плеера;

bottom — задает расстояние от нижней границы элемента до нижнего края экрана плеера.

Значениями атрибутов являются целые числа, задающие размеры в пикселях.

Не обязательно задавать все атрибуты, достаточно указать только те, которые однозначно определяют расположение элемента. Так, для горизонтального позиционирования достаточно указать только left и width, или width и right, или left и right.

Значения для атрибутов right и bottom задаются несколько нестандартно. Обычно все координаты принято отсчитывать от верхнего левого угла области. Но используемый способ задания этих атрибутов удобен, если требуется разместить элемент у правой или нижней границ экрана, независимо от размера экрана.

Следующая группа атрибутов определяет время появления элемента на экране плеера.

begin — атрибут определяет время появления элемента на экране. Если тег <extelm> размещен внутри клипа (внутри тега <clip></clip), то его значением является тайм-код от начала клипа. Если тег размещен внутри тега, определяющего видеофрагмент (<video> или <par>), то он определяет тайм-код относительно начала видеофайла, из которого взят фрагмент.

fragmBegin — атрибут определяет время появления элемента на экране. Этот атрибут указывается только для элементов, размещенных внутри фрагментов. Его значением является тайм-код от начала воспроизведения фрагмента.

dur — атрибут определяет продолжительность нахождения элемента на экране.

При работе с фрагментами иногда бывает удобно определять время появления дополнительного элемента по тайм-коду с начала файла, а иногда от точки воспроизведения фрагмента. Например, если дополнительный элемент привязан к какой-нибудь сцене в фильме, то удобнее его определить по тайм-коду от начала фильма. Как бы потом не было определено начало фрагмента, дополнительный элемент все равно появится в нужный момент.

Пример:

```
<video stream="film1" begin="5:21" dur="90" >
    <extelm begin="5:40" dur="20" … другие атрибуты …. />
</video>
```

Если потом будет изменено определение фрагмента, например на begin="5:05", то время появления дополнительного элемента все равно останется привязанным к сцене с тайм-кодом "5:40" в видеофайле.

Иногда нужно обеспечить появление дополнительного элемента ко времени начала воспроизведения фрагмента. Например, через 5 секунд после начала фрагмента показать название фильма из которого он взят. В этом случае надо использовать атрибут fragmBegin.

Пример:

```
<video stream="seafilm" begin="3:10" dur="80" >
      <extelm fragmBegin="5" dur="10" text="Фрагмент из фильма 'Море'"
            … другие атрибуты …./>
</video>
```

Если не задан ни атрибут begin, ни атрибут fragmBegin, то такой элемент не связан ни с каким тайм-кодом. Но он может быть выведен на экран в какой-то момент времени при активизации другого дополнительного элемента. Активизации дополнительных элементов будет рассмотрена ниже.

Атрибут pause используется, когда нужно остановить воспроизведение при появлении дополнительного элемента. Он должен иметь любое отличное от нуля значение. Например, pause="1" или pause="yes" или pause="true".

Для продолжения воспроизведения необходимо нажать кнопку PLAY на панели управления плеером.

Как было сказано выше, дополнительные элементы могут содержать текстовую или графическую информацию. Для указания типа элемента используются атрибуты text, src и fromId.

Если указан атрибут text, то дополнительный элемент будет текстовым, и строка текста, указанная в качестве значения атрибута, будет выведена в области элемента на экране плеера.

Если указан атрибут src, то элемент является графическим, и изображение, расположенное по URL-адресу, указанному как значение атрибута, будет выведено в области элемента на экране плеера.

Атрибут fromId может быть использован, если плеер размещен на заранее подготовленной странице. Его значением является имя какого-либо HTML элемента (может быть пока невидимого) на странице, заданное атрибутом id. В момент появления дополнительного элемента на экране в область элемента будет перенесена вся информация из указанного элемента HTML страницы.

Если не заданы атрибуты text, src или fromId, то никакая информация в области дополнительного элемента не выводится, но, тем не менее, дополнительный элемент существует как область на экране. С этим элементом

может быть связана некоторая активность, или она может быть графически оформлена (например, задан фон, обведены границы).

Примеры:

```
<extelm left="20" right="20" bottom="5" height="18"
    fragmBegin="5" dur="10" text="Фрагмент из фильма 'Море'" />
```

В полосе, отстоящей от левой и правой границ окна плеера на 20 пикселей и на 5 пикселей от нижней границы, будет выведен текст "Фрагмент из фильма 'Море'"

```
<extelm right="10" top="10" width="100" height="60"
begin="4:44" dur="10" src="./photos/volk03.jpg" />
```

В правом верхнем углу экрана, отступив 10 пикселей от границ, будет выведено изображение, заданное адресом "./photos/volk03.jpg".

Пусть где-то на HTML странице определен элемент

```
<div id="form1" style="display:none;">
    Введите e-mail <br>
    <input id="email" type="text" name="Mail" value=""/><br/>
    <input type="image name=BOK value="Registr" src="./imgs/bokey.gif"
        onClick="Registr()" />
</div>
```

Этот элемент невидим, что определяется атрибутом style элемента. В элементе расположена строка "Введите e-mail", поле для ввода адреса и графическая кнопка для выполнения каких то действий после ввода адреса.

Дополнительный элемент

```
<extelm left="40" right="40" top="30" bottom="30"
    fragmBegin="50" dur="2" pause="yes" fromId="form1" />
```

в момент появления на экране выведет все содержимое этого HTML элемента на экран плеера. При этом воспроизведение будет приостановлено, чтобы дать время ввести запрашиваемую информацию и нажать кнопку.

Для оформления внешнего вида дополнительных элементов используются атрибуты style и class. Эти атрибуты полностью аналогичны соответствующим атрибутам элементов HTML.

Атрибут style определяет стиль оформления дополнительного элемента как элемента HTML. Значением атрибута является строка, содержащая стили оформления.

Пример:

```
<extelm left="20" right="20" bottom="5" height="18" fragmBegin="5" dur="10"
    text="Фрагмент из фильма 'Море'"
    style="color:yellow;font-height:14px;background-color:gray;opacity:0.5;
    text-align:center;" />
```

Атрибут style задает вывод текста по центру дополнительного элемента, высота символов — 14 пикселей, цвет текста — желтый, цвет фона — серый, дополнительный элемент будет иметь 50% прозрачности.

Атрибут style можно использовать для оформления "пустых" дополнительных элементов, у которых не задано содержимое атрибутами text или src.

Пример:

```
<extelm left="220" top="120" width="25" height="25"
    fragmBegin="5" dur="10" style="border:white solid 2px;" />
```

Элемент будет выведен как прямоугольник с белыми границами толщиной 2 пикселя.

Атрибут class используется для оформления внешнего вида дополнительных элементов с использованием каскадных таблиц стилей (CSS). Для его использования необходимо, чтобы плеер находился на заранее подготовленной HTML странице. На этой странице (или в дополнительном файле CSS) должны быть размещены описания классов стилей для дополнительных элементов. Значением атрибута class является имя класса стиля.

Если необходимо создать много одинаково оформленных дополнительных элементов, то предпочтительней создать класс стиля, и на него могут ссылаться все эти дополнительные элементы.

Пример:

На HTML странице (или во внешнем CSS файле) размещены описания классов стилей:

```
<style type="text/css" >
    .subtit {
        color:white;
```

```
            font-family: arial, sans-serif;
            font-height:14px;
            background-color: #1F1F1F;
            opacity:0.5;
            text-align:center;
        }
</style>
```

Тогда дополнительный элемент может ссылаться на этот класс стиля

```
<extelm left="20" right="20" bottom="5" height="18" fragmBegin="5" dur="10"
    class="subtit" text="Фрагмент из фильма 'Море'" />
```

Любой дополнительный элемент может быть интерактивным, то есть он может выполнять некоторые заданные действия в ответ на действия пользователя над этим элементом. Действия пользователя, на которые отвечает элемент, могут представлять собой, например, перемещение курсора мыши в область или из области элемента на экране или щелчок мышью над элементом.

Для придания дополнительному элементу интерактивности используется тег <action>, который размещается внутри тега <extelm> </extelm>. Тег <action> имеет много атрибутов, определяющих, какие действия и при каких событиях должен выполнить плеер.

Атрибут event определяет, при каком событии будет выполняться действие. Он может иметь следующие значения:

— "click" — действие выполняется при щелчке мышью на элементе; это значение предполагается по умолчанию, если атрибут не задан;

— "mouseEnter" — действие выполняется, когда курсор мыши приходит в область элемента;

— "mouseLeave" — действие выполняется, когда курсор мыши покидает область элемента;

— "timeCode" — это значение задает реакцию не на действие пользователя, а на сам факт появления дополнительного элемента на экране в момент, заданный тайм-кодом.

Атрибут srcStage определяет состояние, в которое перейдет плеер перед выполнением действия. Он может принимать следующие значения:

— "play" — продолжить просмотр клипа;

— "pause" — перевести в режим паузы, для продолжения просмотра необходимо нажать кнопку PLAY на панели управления плеером;

— "stop" — прекратить просмотр.

Если атрибут не задан, то сохраняется тот режим, который был перед наступлением события.

Атрибут target определяет, над чем будет выполняться действие. Он может принимать следующие значения:

— "идент.элем." — идентификатор элемента страницы, заданный атрибутом id;

— "имя.элем." — имя дополнительного элемента, заданное атрибутом name;

— "_blank" — будет открыто новое окно браузера.

Далее описываются атрибуты, определяющие действия, которые выполнит плеер:

href — задает url-адрес изображения или HTML страницы, который будет использоваться для изменения элемента страницы;

fromId — задает идентификатор элемента HTML страницы, информация из которой будет использоваться для изменения элемента страницы или дополнительного элемента;

call — задает строку вызова JavaScript функции, определенной на HTML странице; строка должна содержать имя функции и заключенный в скобки список параметров, параметрами могут быть только числовые или строковые константы;

showElms — задает список дополнительных элементов, которые необходимо отобразить на экране; список содержит имена элементов, разделенные пробелами;

hideElms — задает список дополнительных элементов, которые необходимо убрать с экрана; список содержит имена элементов, разделенные пробелами.

clip, fragm, begin — задают имя клипа, имя фрагмента и тайм-код соответственно, на которые необходимо перейти, прервав текущий просмотр.

Место, на которое будет осуществлен переход, определяется тем, какие из этих атрибутов заданы. Если указан атрибут clip, то переход осуществляется в указанный клип, в противном случае переход осуществляется в пределах текущего клипа. Если указан атрибут fragm, то переход осуществляется на указанный фрагмент, а атрибут begin указывает на тайм-код от начала фрагмента. Если атрибут fragm не указан, то атрибут begin определяет тайм-код от начала клипа. Если не указан атрибут begin, то предполагается нулевой тайм-код.

Если переход осуществляется на другой клип, то пользователь может вернуться в прерванное место текущего просмотра, нажав кнопку STOP на панели управления плеером.

destStage — задает в каком режиме будет плеер после перехода на другой просмотр. Он может принимать значения:

"pause" — плеер перейдет в режим паузы; для начала нового просмотра требуется нажать кнопку на панели управления плеером;

"play" — плеер запустит просмотр сразу после перехода.

Если атрибут не задан, то плеер будет находиться в том же режиме, в каком был до наступления события.

Для выполнения различных действий требуется задавать разные комбинации атрибутов тега <action>.

Чтобы выполнить переход на просмотр с другого места, необходимо задать атрибуты clip, fragm, begin и destStage (или некоторые из них).

Примеры:

```
<extelm text=" Вы можете посмотреть дополнительную информацию."
        begin="4:30" dur="15" left="20" right="20" height="16" bottom="10"/>
        <action clip="dopclip" fragm="fish" />
</extelm>

<extelm width="150" height="16" right="30" bottom="20"
      fragmBegin="4" dur="3" pause="yes" text="Пропустить этот сюжет" />
        <action fragm="fragm4"/>
</extelm>
```

Чтобы изменить элемент на HTML странице, необходимо задать его идентификатор в атрибуте target. Если изменяемым элементом является графическое изображение (<IMG>), то в атрибуте href нужно указать URL адрес нового изображения. Если изменяемым элементом является элемент <IFRAME>, то в атрибуте href задается URL адрес новой HTML страницы, которая будет выведена в этом элементе.

Примеры:

На HTML странице имеются элементы

```
<IMG id=photoa src="/images/none.png" width="640" height="480">
<IFRAME id="map" whidth="600" height="600">
```

Интерактивные дополнительные элементы могут изменить содержимое этих элементов

```
<extelm width="200" height="150" right="10" top="10" begin=6:43" dur="15"
      src="/photos/smal/artist1.jpg" >
```

```
    <action target="photoa" href="/photos/large/artist1.jpg" />
</extelm>
```

При нажатии на фотографию, являющуюся дополнительным элементом, на странице появится увеличенное изображение в заранее подготовленном элементе IMG

```
<extelm left="254" top="180" width="200" height="15" begin="23:18" dur="5"
      text="Посмотреть карту" pause="yes">
      <action target="map"
            href="www.mapservice.org?cx=12345.6789&cy=78901.23456"/>
</extelm>
```

При нажатии на текст дополнительного элемента, в элементе IFRAME появится страница с сайта указанного картографического сервиса.

Чтобы открыть новое окно браузера атрибут target должен быть задан как "_blank", а атрибут href — задавать URL адрес открываемой страницы.

Пример:

```
<extelm left="254" top="180" width="200" height="15" begin="23:18" dur="30"
      text="Посмотрите сайт музея" >
      <action target="_blank" href="www.histor-museum.org" srcStage="pause"/>
</extelm>
```

Чтобы изменить элемент на HTML странице или дополнительный элемент на экране плеера, можно переместить в них информацию из другого, может быть невидимого, элемента HTML страницы. Для этого в атрибуте target должен быть задан идентификатор изменяемого элемента страницы или имя дополнительного элемента, а атрибут fromId долженсодержать идентификатор элемента страницы, из которого информация будет перенесена в изменяемый элемент.

Пример:

На экране плеера есть два элемента

```
<extelm name="info" left="5" top="5" width="300" height="200"
    begin="23:18" dur="30"/>

<extelm left="10" bottom="10" width="120" height="15" begin="23:18" dur="10"
      text="Показать справку о Петрове">
    <action target="info" fromId="petrov" srcStage="pause"/>
</extelm>
```

а на HTML странице есть невидимый элемент

```
<DIV id="petrov" style="display:none;">
      Петров С.М.<br>
      Родился 31.07.1960 г.ф<br>
      Окончил МГУ в 1983 г.<br>
      Работает в РАН с 1991г.
</DIV>
```

Первый дополнительный элемент первоначально не имеет никакого содержимого, это просто область на экране плеера. При щелчке мыши на тексте второго дополнительного элемента в эту область будет перенесена информация из невидимого элемента DIV с HTML страницы.

Чтобы вывести на экран плеера дополнительные элементы или убрать с экрана некоторые элементы необходимо использовать атрибуты showElm и hideElm. Атрибут showElm задает список имен выводимых дополнительных элементов, а атрибут hideElm — список убираемых. Дополнительные элементы, которые можно интерактивно выводить и убирать, обычно не связаны с тайм-кодами.

Пример:

```
<extelm left="20" bottom="10" width="200" height="15" begin="23:18" dur="30"
      text="Кто есть кто" >
      <action showElm="elIvanov elPetrov elSidorov"/>
      <action event="mouseLeave" hideElm="elIvanov elPetrov elSidorov"/>
</extelm>
<extelm name="elPetrov" left="20" ="256" width="80" height="15"
      text="Петров"/>
<extelm name="elIvanov" left="180" ="96" width="80" height="15"
      text="Иванов"/>
<extelm name="elSidorov" left="320" ="175" width="80" height="15"
      text="Сидоров"/>
```

При щелчке мыши на элементе с текстом "Кто есть кто" на экран плеера будут выведены дополнительные элементы с фамилиями людей. Когда курсор мыши уйдет из области элемента с текстом, ранее выведенные элементы будут убраны с экрана.

Интерпретация описанных в настоящей статье управляющих структур на языке XML, предназначенных для поддержки технологии гипервидео в среде HTML5, была в основном реализована в рамках пилотной версии плеера.

Предполагается в дальнейшем разработать интерактивную клиентскую программу, способную функционировать на мобильных устройствах, используя возможности глобальной навигации. В этой версии плеера текущие координаты устройства также смогут быть объявлены как управляющие события.

Литература

1. Бухштаб Ю.А., Воробьев А.А., Евтеева Н.Н. Методы виртуального редактирования в распределенной среде видео и аудио потоков, представленных в различных форматах // Препринты ИПМ им. М.В.Келдыша. 2011. №67. 12 с. URL: http://library.keldysh.ru/preprint.asp?id=2011-67

2. 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. 15 с. URL: http://library.keldysh.ru/preprint.asp?id=2012-62

3. HTML5 — A vocabulary and assotiated APIs for HTML and XHTML. — W3C Working Draft 25 October 2012. — URL: http://www.w3.org/TR/html5/.