

1. Введение.

С приходом в нашу жизнь сети Интернет и высокоскоростных сетевых каналов возникли технические возможности доступа к медиа объектам с домашних, школьных и рабочих компьютеров. Оцифровка и хранение большого количества мультимедийных объектов, в том числе и видеoinформации, становятся все более распространенной практикой. Появляются все новые технологии, позволяющие использовать мультимедийные объекты.

Такие компании как Real Networks, Microsoft и Apple Computer разработали новые технологии потокового видео, т.е. технологии, когда видео объекты не загружаются, а «транслируются» по сети Интернет или локальной сети, обеспечивая возможность их просмотра. Однако базовое программное обеспечение не позволяет пользователю манипулировать такими объектами, осуществляя полнофункциональное редактирование видео, размещенного на видеосервере, создавая таким образом свои собственные потоковые видеопоследовательности.

В рамках создания инструментального программного комплекса, предоставляющего возможность использования визуальных данных различных типов, в том числе и потокового видео, авторами данной статьи разрабатываются программные средства, обеспечивающие интерактивное редактирование потоковой видеoinформации.

Предполагается, что в целом инструментальный программный комплекс будет интегрировать в единой среде следующие возможности:

- использование поискового механизма, опирающегося как на индексирование метаинформации, ассоциированной с изображениями и видеофильмами, так и на содержательный поиск с использованием визуальных примитивов (характеристик изображения, которые автоматически вычисляются по оцифрованным визуальным данным);
- регистрацию цифруемой видеoinформации и генерацию представительных стоп-кадров;
- организацию тематических информационных подпространств (совокупности базовых объектов разных типов и присоединенных процедур, объединенных общей проблематикой и позволяющих обеспечить полную информационную поддержку конкретному пользователю или группе пользователей);
- просмотр потокового видео с использованием таких средств, как остановка, покадровый просмотр, перемотка и т.д.;

- отбор в ходе просмотра релевантных для пользователя видеофрагментов и их интерактивный монтаж с целью автоматического создания целевой видеопоследовательности и ее характеристик;
- обеспечение доставки пользователю смонтированных им видеоклипов в потоковом режиме;

Авторы настоящей статьи на протяжении нескольких лет занимались проблематикой организации управления базами визуальных данных, и в рамках этих исследований были разработаны и используются на практике программные средства, обеспечивающие обработку метаданных и поисковых образов, отражающих визуальное содержание изображений и видеоинформации. Однако, наиболее сложной компонентой рассматриваемого программного комплекса являются программы, обеспечивающие реализацию работы с потоковым видео, при этом прежде всего речь идет о средствах, способных обеспечить интерактивный монтаж в среде Internet или Intranet видеоинформации расположенной на сервере. Вместе с тем возможности такого удаленного монтажа позволяют решать ряд практических задач, возникающих в различных предметных областях. Рассмотрим теперь некоторые примеры приложений программных инструментальных средств, реализующих интерактивный видеомонтаж потокового видео.

Так, эти программные средства могут применяться как одна из компонент систем обеспечения принятия управляющих решений. Действительно, задачи связанные с проблематикой управления общественными и производственными процессами являются весьма сложными, и персонал, непосредственно осуществляющий принятие решений, нуждается в адекватной информационной поддержке. В частности, все большее значение приобретает возможность оперативного доступа к визуальной информации, в том числе видеоинформации. В настоящее время во многих государственных учреждениях, крупных предприятиях и т. д. уже созданы собственные видеоархивы, а иногда имеется возможность получать видео с места событий в реальном времени. Однако не всегда персонал, ответственный за принятие решений, в состоянии эффективно использовать эту информацию. На базе предлагаемых инструментальных средств можно организовать доступ к видеоинформации на новом, качественно более высоком, уровне. Сотрудники смогут просматривать потоковое видео, передаваемое с сервера видеоархива по локальной сети или через Интернет, на своих рабочих местах, отбирая нужные им фрагменты и осуществлять монтаж этих фрагментов, формируя таким образом нужную им видеопоследовательность.

В сфере образования рассматриваемые программные средства позволяют создавать и использовать учебные on-line курсы и образовательные энциклопедии, представленные в виде гипертекстов, сопровождаемые объемными видеоклипами. При этом, с одной стороны, преподаватель сможет использовать рассматриваемый программный инструментарий для подготовки собственных спецкурсов, осуществляя монтаж нужных ему фрагментов видеофильмов, транслируемых в режиме потокового видео, и далее комбинировать их со своими текстами, создавая таким образом свои собственные мультимедийные учебные курсы. С другой стороны, студенты смогут использовать эти же интерактивные возможности для подготовки курсовых работ. Такой подход позволит предоставить прямо в классных комнатах возможность не только просматривать документальные и учебные фильмы, но и получить возможность активно использовать их, что безусловно инициирует реальную заинтересованность изучаемым предметом. Так, например, при изучении истории возможность для студентов увидеть своими глазами реалии прошедших событий, лица участников этих событий и т.д. сделает обучение не механическим процессом запоминания имен и дат, но скорее самостоятельным творческим исследованием.

Еще одной областью возможных приложений рассматриваемых программных средств является кинопроизводство. В процессе создания документальных фильмов одним из самых важных этапов является подбор нужных сюжетов в видеоархивах. При этом нужный архив часто находится не только в другом городе, чем производящая студия, но и в другой стране. Предполагается, что режиссер фильма будет через Интернет осуществлять поиск интересующих его фильмов в базе данных видеоархива, содержащей текстовые описания фильмов и представительные стоп-кадры. Затем он будет просматривать отобранные им фильмы в режиме потокового видео без переписывания видеофайлов с сервера, отмечая тайм-коды нужных ему фрагментов. При этом будет автоматически подготовлен список нужных ему видеоматериалов с данными оригинала. Далее этот список может быть использован для заказа видеофрагментов профессионального качества.

2. Принципы построения программного обеспечения.

Как уже упоминалось, в настоящий момент для работы с потоковым видео практически используются три программных продукта компаний Real Networks, Microsoft и Apple Computer. Принципы функционирования этих продуктов, вообще говоря, одни и те же. Видео информация сначала

значительно сжимается с помощью кодировщика и преобразуется в пригодный для потоковой передачи формат, далее размещается на видеосervere, а затем передается пользователям в виде непрерывного потока (в принципе видеоинформация может кодироваться в реальном времени и таким образом осуществляться прямое вещание через видеосerver). Однако, каждый из этих программных продуктов использует свой собственный формат. В ходе проведенных исследований рассматривались их характеристики, и затем они были опробованы в работе. Также был проведен ряд программных экспериментов, нацеленных на оценку возможности разработки средств, предназначенных для поддержки интерактивного видеомонтажа. В результате было принято решение разрабатывать описываемый инструментальный комплекс на базе программных средств от RealNetworks.

В ходе проведенных работ были спроектированы программные средства, обеспечивающих возможность интерактивного видеомонтажа, и возможность их реализации была подтверждена в ходе испытаний соответствующей программной модели.

Рассмотрим теперь общие принципы построения этих программных средств. Предполагается, что на видеосervere Helix Universal Server хранится набор видеофильмов, закодированных в виде rm-файлов. Ссылки на эти файлы и метаданные, описывающие соответствующие фильмы, содержатся в специализированной базе данных на HTTP-сервере. Используя соответствующий поисковый механизм пользователь может выбрать нужный ему фильм и вызвать его на просмотр. Далее он может просматривать фильм в модифицированном окне RealPlayer. При этом пользователь имеет возможность выполнять такие действия как остановку и возобновление просмотра клипа, его пошаговый просмотр (как вперед, так и назад) с выбором величины шага начиная от 0,1 секунды (тесты показали, что позиционирование с такой точностью вполне возможно), изменение размера окна RealPlayer и т.д. Управление просмотром осуществляется программами, написанными на JavaScript и находящимися на компьютере пользователя (эти программы взаимодействуют с RealPlayer, а тот, в свою очередь, обращается к Helix -серверу).

Во время просмотра пользователь может отмечать начало и конец интересующих его фрагментов фильма. Эти отметки (тайм-коды) обрабатываются соответствующей программой и передаются на сервер. При этом пользователь указывает как должен быть обработан этот фрагмент, например, хочет ли он выкинуть его из фильма или наоборот объявить его новым - «пользовательским» фильмом (клипом). Информация

о выбранных пользователем фрагментах и способах их обработки с помощью специальных CGI-программ запоминается сервером (HTTP-сервером) в базе данных. Фактически, на видеосервере хранятся в неизменном виде исходные фильмы, а «пользовательские» клипы описываются в базе данных соответствующим URL и тайм-кодами начала и конца.

Кроме того, соответствующие CGI-программы осуществляют действия по редактированию клипов. При этом пользователю предоставляется возможность объединять фрагменты из различных фильмов (не важно исходных или «пользовательских») в нужной последовательности, создавая таким образом свои новые клипы, которые представляются в базе данных многоуровневыми структурами соответствующих URL и тайм-кодов. Эти программы осуществляют генерацию SMIL-файлов, содержащих описание выбранных пользователем фрагментов создаваемого нового клипа. Генерируемые SMIL-файлы представлены в формате XML и задают информацию о воспроизводимых под их управлением клипах, прежде всего информацию о начальных и конечных тайм-кодах выбранных фрагментов. Кроме того, в SMIL-файлах задается последовательность клипов. Под управлением этой информации RealPlayer производит упреждающее считывание фрагмента, и испытания созданной программной модели показали, что переход с клипа на клип происходит без видимой временной задержки. Также было установлено, что SMIL-файлы позволяют использовать фрагменты из клипов, записанных с различными разрешениями и с различными скоростями. Такие фрагменты можно объединить в единый клип, что в принципе позволяет использовать фрагменты из клипов, находящихся на разных серверах, указывая в SMIL-файлах соответствующие URL-адреса подобранных пользователем фрагментов. Таким образом, предлагаемая программная модель видеоредактора позволяет пользователю удалять и выбирать фрагменты из исходных фильмов, добавлять выбранные фрагменты в конец создаваемого им нового клипа или вставлять его в любое место этого клипа. Действия пользователя по редактированию видеоданных передаются в интерактивном режиме серверу, который вызывает соответствующие CGI-программы, которые модифицируют базу данных и редактируют SMIL-файлы. Результаты своей работы пользователь может тут же просматривать в окне RealPlayer. Таким образом, создаваемый пользователем клип представляется только SMIL-файлом, а не реальным видеофайлом.

В дальнейшем этот созданный пользователем новый клип может использоваться таким же образом как исходный клип. Из него также можно выбирать фрагменты, поскольку RealPlayer воспроизводит клип, заданный

SMIL-файлом, как единый фильм, производя отсчет тайм-кодов от начала первого фрагмента.

3. Реализация.

Как уже было сказано, для обработки информации на сервере используется специальная CGI-программа, вызываемая по запросу броузера. Броузер передает программе информацию из обычных и скрытых полей форм, содержащихся на WEB странице. Эта информация указывает какие действия и с какими параметрами необходимо выполнить.

CGI-программа выполняет следующие функции:

- Поиск документов в базе данных.
- Создание пользовательского клипа.
- Вставка фрагмента в клип или удаление фрагмента из клипа.
- Удаление пользовательского клипа.

Все выше перечисленные действия связаны с обращением и/или модификацией базы данных. Кроме того, CGI-программа генерирует временный SMIL файл для клипов, состоящих из нескольких фрагментов, и передает его в RealPlayer.

Все действия, связанные с просмотром клипа и выделением фрагментов, реализованы не с помощью CGI-программы, а с помощью JavaScript и выполняются непосредственно в броузере.

Параметры, передаваемые в CGI-программу, имеют вид:

Имя_параметра=Значение

Как принято в протоколе HTTP несколько параметров соединяются знаком &, а пробелы кодируются знаком +

Например:

DO=SHOWC - Показать список исходных клипов.

DO=SHOWF&USER=Film1 - Показать список клипов пользователя Film1.

DO=MAKEF&USER=Film1 TITLE=Мой+клип - Создать новый клип пользователя Film1 с названием «Мой клип».

DO=INSFR&IDCL=10056&TIMEINS=32300&IDCLINS=200356&TIMEB=10000&

TIMEE=25500 - Вставить в клип 10056, в точку с таймкодом 32,3 сек. фрагмент из клипа 200356, с тайм-кодами начала и конца 10 сек и 25,5 сек.

Каждый клип в базе данных имеет свой уникальный номер, и ссылка на клип делается по этому номеру.

Результатом работы CGI-программы, кроме модификации базы данных и создания SMIL файлов, обязательно является HTML документ, который отображается в соответствующем фрейме. Для генерации HTML документов, используется специальный файл, содержащий параметризованные заготовки отдельных частей документа. Это позволяет менять форму выдачи, не меняя CGI-программы. В качестве параметров могут использоваться как поля базы данных, так и специальные переменные CGI-программы.

Рассмотрим теперь пример параметризованного задания HTML документа для выдачи списка клипов базы данных и кнопок выбора (Здесь по правилам HTML комментарии содержатся между символами <!-- и -->).

```
###TIT
<!-- Начало HTML страницы -->
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=windows-1251" lang="ru">
<LINK href="http://vor/vtest/snap.css" type=text/css rel=stylesheet>
</head>
<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<div class="txt">
```

```
###LISTB
<!-- Начало списка документов -->
<form name="f1">
```

```
###LISTD
<!-- Генерация документа. Параметры [NREC] - номер клипа, [TIT] -
название клипа. -->
<!-- параметры NREC и TIT берутся из полей базы данных. -->
<!-- Такая строка генерируется для каждой выводимой записи. -->
<input type=radio name=clip value="[NREC]"> [TIT] <br>
```

```
###LISTE
```

```
<!-- Конец списка документов -->
```

```
</form>
```

```
###FIN
```

```
<!-- Конец HTML страницы -->
```

```
</div>
```

```
</body>
```

```
</html>
```

CGI-программа может вывести следующий HTML документ.

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
```

```
<html>
```

```
<head>
```

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;  
charset=windows-1251" lang="ru">
```

```
<LINK href="http://vor/vtest/snap.css" type=text/css rel=stylesheet>
```

```
</head>
```

```
<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
```

```
<div class="txt">
```

```
<form name="f1">
```

```
<input type=radio name=clip value="100001"> Клип-1. <br>
```

```
<input type=radio name=clip value="100002"> Клип-2. <br>
```

```
<input type=radio name=clip value="100003"> Клип-3. <br>
```

```
<input type=radio name=clip value="100004"> Клип-4. <br>
```

```
<input type=radio name=clip value="100005"> Клип-5. <br>
```

```
<input type=radio name=clip value="100006"> Клип-6. <br>
```

```
<input type=radio name=clip value="100007"> Клип-7. <br>
```

```
<input type=radio name=clip value="100008"> Клип-8. <br>
```

```
</form>
```

```
</div>
```

```
</body>
```

```
</html>
```

Точно также формируется SMIL файл для клипов, состоящих из нескольких фрагментов. Параметризованная информация для SMIL файла следующая:

```
###SMILB
```

```

<!-- Начало SMIL файла-->
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
<head>
<layout>
<!-- Описание области вывода -->
<region id="video_region" width="360" height="270" z-index="1" fit="fill"
backgroundColor="black"/>
</layout>
</head>
<body>
<seq>

```

###SMILD1

```

<!-- Этот фрагмент выводится для одиночного фрагмента -->
<!-- вместо параметров $CLIPURL, $SMILTB $SMILTE программа
додставляет значения -->
<!-- URL фрагмента, начальный и конечный таймкоды из специальных
переменных -->
<video src="[$CLIPURL]" region="video_region" clipBegin="[$SMILTB]s"
clipEnd="[$SMILTE]s" fill="freeze"/>

```

###SMILDP1

```

<!-- Этот фрагмент выводится для фрагмента, вместе с которым идет
предзагрузка следующего -->
<!-- Параметр [$NCLIP] указывает номер параллельного блока для
синхронизации. Этот параметр используется как в операторе <par>, так и в
операторе <video>, таким образом параллельный блок закончится, когда
закончится показ видеофрагмента-->
<par endsync="c[$NCLIP]">
    <video src="[$CLIPURL]" id="c[$NCLIP]" region="video_region"
clipBegin="[$SMILTB]s" clipEnd="[$SMILTE]s" fill="freeze"/>

```

###SMILDP

```

<!-- Этот фрагмент выводится для предзагрузки фрагмента, и его показа в
следующем параллельном блоке -->
    <prefetch src="[$CLIPURL]" bandwidth="30000"/>
</par>
<par endsync="c[$NCLIP]">
    <video src="[$CLIPURL]" id="c[$NCLIP]" region="video_region"
clipBegin="[$SMILTB]s" clipEnd="[$SMILTE]s" fill="freeze"/>

```

###SMILDPL

<!-- Этот фрагмент выводится для предзагрузки последнего фрагмента, и его показа -->

```
<prefetch src="[$CLIPURL]" bandwidth="30000"/>
```

```
</par>
```

```
<video src="[$CLIPURL]" region="video_region" clipBegin="[$SMILTB]s"
clipEnd="[$SMILTE]s" fill="freeze"/>
```

###SMILE

<!-- Конец SMIL файла -->

```
</seq>
```

```
</body>
```

```
</smil>
```

В результате может быть сгенерирован файл следующего содержания:

<!-- Начало SMIL файла -->

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
```

```
<head>
```

```
<layout>
```

<!-- Описание области вывода -->

```
<region id="video_region" width="360" height="270" z-index="1" fit="fill"
backgroundColor="black"/>
```

```
</layout>
```

```
</head>
```

```
<body>
```

<!-- Начало последовательного исполнения -->

```
<seq>
```

<!-- Начало параллельного исполнения -->

<!-- Показ клипа gagarin2.rm от тайм-кода 18.5 с до тайм-кода 20 с и предзагрузка клипа leonov.rm -->

```
<par endsync="c1">
```

```
<video src="rtsp://vor/vtest/gagarin2.rm" id="c1" region="video_region"
clipBegin="18.5s" clipEnd="20.0s" fill="freeze"/>
```

```
<prefetch src="rtsp://vor/vtest/leonov.rm" bandwidth="30000"/>
```

```
</par>
```

<!-- Начало параллельного исполнения -->

<!-- Показ клипа leonov.rm от таймкода 3.8 с до таймкода 7.2 с и предзагрузка клипа c0304.rm -->

```

<par endsync="c2">
  <video src="rtsp://vor/vtest/leonov.rm" id="c2" region="video_region"
clipBegin="03.8s" clipEnd="07.2s" fill="freeze"/>
  <prefetch src="rtsp://vor/vtest/c0304.rm" bandwidth="30000"/>
</par>
<!-- Начало параллельного исполнения -->
<!-- Показ клипа c0304.rm от таймкода 9.5 с до таймкода 13.9 с и
предзагрузка клипа video1.rm -->
<par endsync="c3">
  <video src="rtsp://vor/vtest/leonov.rm" id="c3" region="video_region"
clipBegin="09.5s" clipEnd="13.9s" fill="freeze"/>
  <prefetch src="rtsp://vor/vtest/video1.rm" bandwidth="30000"/>
</par>
<!-- Показ клипа video1.rm от таймкода 0 с до таймкода 3.9 с -->
<video src="rtsp://vor/vtest/video1.rm" region="video_region"
clipBegin="00.0s" clipEnd="03.9s" fill="freeze"/>
</seq>
</body>
</smil>

```

Этот файл не передается браузеру. Он записывается в специальную директорию, а затем его URL передается в RealPlayer.