User adaptable multimedia presentations for the World Wide Web

Franck Rousseau [*,1], J. Antonio García-Macías [1], José Valdeni de Lima [1], Andrzej Duda [1]

LSR–IMAG Laboratory, B.P. 72, 38402 Saint Martin d'Hères, France

Abstract

We propose a generic solution for user adaptation of synchronized multimedia presentations. We consider adaptation as a transformation problem: the user specifies a predicate that applied to a generic multimedia presentation yields a customized view of the presentation. We specify a means of expressing content descriptions and alternate content in generic multimedia presentations. User adaptation is based on content predicates that a player uses to select alternate content or elements that match content descriptions. Several examples show flexibility and expressing power of the proposed approach. © 1999 Published by Elsevier Science B.V. All rights reserved.

Keywords: Synchronized multimedia; User adaptation; Accessibility issues; Metadata; Navigation techniques

1. Introduction

The existing World Wide Web infrastructure provides access to a rich information space of hypermedia documents that integrate media types (JPEG, GIF, MPEG, audio files). For various reasons, we want to be able to *adapt* World Wide Web documents to different conditions and user needs:

- *End platform:* a document may need to be adapted to the characteristics of an execution platform: screen size, colour depth, processing power.
- *Network connectivity:* available bandwidth and packet loss rate may influence the encoding and compression ratio of multimedia data.
- *Accessibility:* a document may need to be adapted to different accessibility constraints; a browser may present alternate content to disabled individuals or change media type in some circumstances (e.g. World Wide Web access when driving).

- *User needs:* the user may want a personalized version of a document. Customizing may depend on various constraints:
  - *user profile:* a document should be adapted according to some user profile;
  - *education level:* an appropriate version of a course or a lesson should be tailored to a required level of education;
  - *time:* the user may require a short version or a summary of a document; in this case, the short version or the summary should include different parts of a document depending on the user's focus or interest;
  - *access history:* different content may be presented depending on the history of access (e.g. a given access pattern results in a customized version).

All these constraints expressed by the user can be considered as a *predicate* that the user specifies to create a personalized view of a document.

Several different means already exist for adapting World Wide Web documents.

* Corresponding author.
[1] {rousseau,amacias,valdeni,duda}@imag.fr

- *Content negotiation* [6,22]: HTTP/1.1 allows a client to access the 'best' version of a document, if several versions exist on a server ('best' means that the version best fits the user needs specified in the request header).
- *Web Access Initiative (WAI)* [27]: addresses making the Web accessible to people with disabilities. WAI already proposes accessibility guidelines for creating documents that can be transformed according to user needs.
- *W3C Mobile Access Activity and Wireless Application Protocol Forum (WAP)* [21]: are dedicated to enabling advanced services and applications on mobile wireless devices (such as cellular telephones). WAP has defined several standards such as the Wireless Markup Language (WML) and Wireless Application Environment (WAE) to deal with specific needs of low-end devices and low-bandwidth network connections.
- *URI:* allows to identify different versions of a document thus enabling a client to choose (via HTTP/1.1) a version that suits the client characteristics the best.
- Some other World Wide Web standards provide useful support for adaptation:
  - *Document Object Model (DOM)* [25]: provides a standard set of objects for representing HTML and XML documents [23], a standard model of how these objects can be combined, and a standard interface for accessing and manipulating them.
  - *Cascading Style Sheets (CSS)* [28]: allow personalized presentation styles of document contents.
  - *Document Style Semantics and Specification Language (DSSSL)* [13]: provides a programming language for document transformations and a query language for document elements.
  - *Resource Description Framework (RDF)* [26]: provides a means of associating semantics with World Wide Web documents.
  - *Query languages for the World Wide Web* [29]: this new activity of W3C tries to leverage existing database query languages and use them in the context of XML documents and RDF descriptions.

Although user adaptation concerns all kinds of documents, we will focus on *synchronized multimedia presentations*. A synchronized multimedia presentation integrates multiple streams of continuous media, and specifies how they are combined together and how they should be presented to the user. W3C has initiated an activity to explore integration of synchronized multimedia into World Wide Web documents. It has issued a recommendation of SMIL (*Synchronized Multimedia Integration Language*) [24], a new specification language with temporal functionalities.

In a previous work [15], we have proposed another approach to the specification of synchronized multimedia presentations based on three concepts:

- *hypertime links* for temporal composition,
- common *time bases* for close lip-sync synchronization between media objects,
- *dynamic layout* that can be seen as an extension of media objects.

Our approach provides a simple and clean way of specifying temporal composition and provides several advantages compared to SMIL. In particular, it allows to separate the temporal composition from the definition of media objects. Another advantage is that we can easily add new elements to deal with adaptable presentations.

The objective of this paper is to propose a generic solution for user adaptation of synchronized multimedia presentations. We use our temporal extensions to HTML as a base for adding adaptation support. As we have seen above, adaptation can be seen as a transformation problem: the user specifies a predicate that, applied to a synchronized multimedia presentation, generates a customized view of the presentation. To do so, we need a means of specifying:

- *alternate content:* elements that allow to define alternate media objects, temporal composition, and layout,
- *content descriptions:* a way of associating metadata information with elements of a presentation (e.g. importance, keywords, duration),
- *content predicates:* a condition that selects alternate content or elements that match content descriptors.

The means should fit the temporal model, so that applying a predicate on a given presentation yields a result which is a valid temporal presentation. Other support for adaptation that we have mentioned (content negotiation, WAI, WAP, CSS), can always

be used to enrich adaptation facilities defined in this paper.

In the remainder of the paper, we give a motivating example (Section 2) that helps understanding the adaptation problem. Then, we present our temporal extensions to HTML (Section 3) and define elements needed for expressing content descriptions, alternate content, and content predicates (Section 4). The use of this adaptation support is shown in an example (Section 5). Finally, we compare our solution with related work (Section 6) and outline conclusions (Section 7).

2. Motivating example

We present an example to illustrate what we want to achieve. Imagine a presentation of a travel agency that offers journeys to different countries. Such a presentation includes video clips, background music, images, and narrative voice. For each journey, places to visit are illustrated and commented. The agency presents its offer and precise conditions. The same scenario repeats for each country. Fig. 1 presents an example of a long presentation and a short summary for two countries: China and Japan.

We would like to be able to adapt the presentation to user preferences. As an example, we consider two cases.

- *Choice of a country:* the user may want to see only the presentation of one country or a given set of countries instead of being forced to follow the whole presentation.
- *Summary:* instead of the whole presentation, the user may want to see a short summary of the offer for all countries. Showing only the first 10 s of each country presentation is probably not the right thing to do. Rather, we would like to see a sequence of 10 s summaries showing the most important things that can be visited in each country. Moreover, less important media objects, such as some images or background music, may be dropped.

The two cases of the adaptation show the characteristics of the problem. We need a means of choosing alternate media objects in a presentation, specifying alternate temporal paths within a global temporal composition, and associating descriptions with objects, so that the user may choose the right contents. In addition to that, we need a way to specify the user choices and preferences.

Obviously, a simple solution to the adaptation problem can be to maintain different versions of the presentation for different usage. However, this means that we know the user's needs and we are able to generate all possible versions depending for example on different languages, formats, and contents. We think that such a solution is cumbersome and we propose another approach to avoid multiplication of versions:
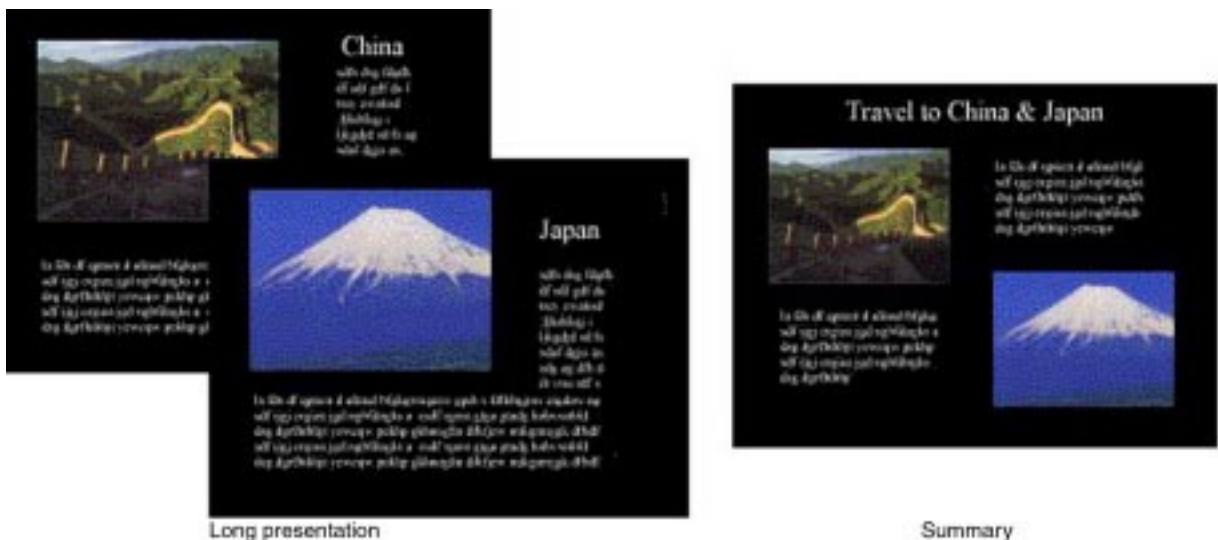

Fig. 1. An example of a presentation.

customization of a generic presentation to user needs.

Our solution is based on the previous work that adds temporal extensions to HTML [15]. To deal with adaptation, we define new elements and attributes to mark up alternate contents and add descriptions. We follow the RDF approach to associate descriptions with media objects and temporal links. A player of a generic presentation will then allow the user to specify a predicate to tailor a given presentation to his/her needs.

The proposed solution is generic in the sense that it can be useful for other purposes than those introduced in the example: to augment accessibility by tailoring presentations to user constraints, to adapt presentations to a specific platform or network constraints (the domain of WAP); and it may also be used in more a general context to query collections of multimedia presentations. All existing support mentioned in the previous section (content negotiation, URI, CSS) is complementary and can be used in conjunction with the proposed solution.

3. Temporal extensions to HTML

Before introducing support for adaptation, we present the extensions to HTML, so that we can il-

lustrate our proposition with a complete motivating example. The extensions are based on the following concepts [15]: *hypertime links*, *time bases*, and *dynamic layout*. Implementation details of a prototype player can be found elsewhere [16].

3.1. Hypertime links for temporal composition

We propose to use a simple functional paradigm derived from temporal point nets to specify temporal composition: a temporal link between an origin and a target. We call it a *hypertime link* by analogy to its World Wide Web companion. A hypertime link has explicit temporal semantics: it relates two media samples (e.g. video frames) and assigns time instants to the samples. Following a hypertime link is automatic in the sense that it does not require any user interaction and consists of skipping in time from the origin of the media sample to the target. The action expressed by the link depends on the target: if the target is the beginning of a media object or a sample somewhere inside the object, the link activates the target. If the target is the end of a media segment, the link terminates the object.

We use the notion of *time points* to define an origin and a target.

```
< htlink orig   = " time-point-id | [ time-point-id, time-point-id ] "
         target = " time-point-id " >
```

The time point tag defines a position in a media object or layout that can be used as an origin or a target of a hypertime link. The position can be of two types: nominal or absolute. A nominal position is defined using the nominal presentation rate of a media object. An absolute position is based on an absolute time coordinate of a time point.

```
< time-point id    = " name "
             value = " integer value | float value "
             unit  = " frame | sample | timestamp | second | ... "
             type  = " nominal | absolute " >
```

3.2. Common time bases for close synchronization

Specification of temporal composition is not sufficient to play back a multimedia document. We need some more information about how media objects must be synchronized. This information is particularly useful in a computing environment that does not provide strong real-time support. In such an environment, different media segments started at the same instant may get out of synchronization after some time and require some corrective action (such as dropping samples) to become synchronized again. We want to be able to specify which objects

should be kept synchronized, how often, and what is the nature of this close synchronization (in other words, who is the master of the time).

For this purpose, we define the notion of a *time base*. A time base is a virtual time space in which media objects 'live'. A time base defines a common time coordinate space for all objects that are related by some relations, for example master–slave dependency. A time base can be seen as a perfect time space in which real-world phenomena such as jitter or drift do not exist and media objects behave in a perfect manner. Obviously, such a perfect time space does not exist; however, it can be implemented closely enough using real-time support. If such support is not available, which is the case in many existing systems, a document should indicate how the quality of presentation is to be maintained and what is the nature of synchronization to be enforced.

We define the nature of synchronization between media segments using the notions of *master* and *slave*. A master–slave relationship defines a master controlling a slave according to its needs. We extend this notion to multiple masters and slaves through the common time base: a master can accelerate time or slow it down, hence slaves and other masters must adjust to time. The master–slave relationship allows the user to easily define the behaviour of media segments with respect to synchronization.

The time base tag groups layouts and media objects that should be synchronized closely. Synchronization roles (master or slave) are specified in objects themselves.

```
< timebase >
      layouts
      objects
< /timebase >
```

A synchronization point can be specified in a media object:

```
< sync-point id    = " name "
             value = " integer value | float value "
             period = " integer value | float value "
             unit  = " frame | sample | timestamp | second | ... "
             type  = " nominal | absolute " >
```

3.3. Media objects and dynamic layout

We suppose that a synchronized multimedia document may include a variety of media objects having temporal behaviour. A *media object* defines time evolution of media samples of one type. Media samples must be presented at precise time instants defined by the rate of presentation. The rate may be imposed by the author, adapted to match the duration of another object, or adjusted to synchronize with other objects. A media object schedules presentation of samples within a given time base. In this way, objects in the same time base are synchronized.

In addition to synchronized presentation of media samples, a media object can be controlled by other objects according to temporal composition. A hypertime link activated by another object can change the current presentation of an object and force it to skip to the target sample or to stop.

The media object tag defines an object that refers to the URL of media content, specifies the role for master or slave synchronization, and defines a scale temporal transformation. Within the tag, we can encapsulate other objects and define synchronization and time points. Unlike SMIL, we do not distinguish between different media objects. The media object tag may contain arbitrary content presented using a given layout.

```
< object id    = " name "
         src   = " url "
```

```
        role  = " master | slave "
        scale = " object-id | layout-id | float value " >
     objects
     synchronization points
     time points
     hypertime links
< /object >
```

We define a *dynamic layout* as a special case of a media object. It defines a temporal behaviour of the physical layout. The only difference is that layouts are neither masters nor slaves since they do not contain any media samples to be synchronized. Frames can include other layouts to specify nested layouts that provide nested coordinate spaces. Hypertime links define how the layout changes in time. This approach allows seamless integration of spatial and temporal dimensions into a multimedia document.

The layout tag is a special case of a media object tag. It does not specify a synchronization relationship nor media content. Instead of encapsulating other objects, it encapsulates *frames*, a means of defining regions of a screen in which media objects may be presented.

```
< layout id    = " name "
          scale = " object-id | layout-id | float value " >
     frames
     synchronization points
     time points
     hypertime links
< /layout >

< frame id   = " name "
        src   = " object-id | layout-id | url "
        layer = " integer value "
        shape = " shape "
        mask  = " mask " >
```

3.4. Motivating example using the extensions

We present below a simplified version of the example described in Section 2 (the layout is not given for the sake of clarity). Fig. 2 shows its graphic representation.
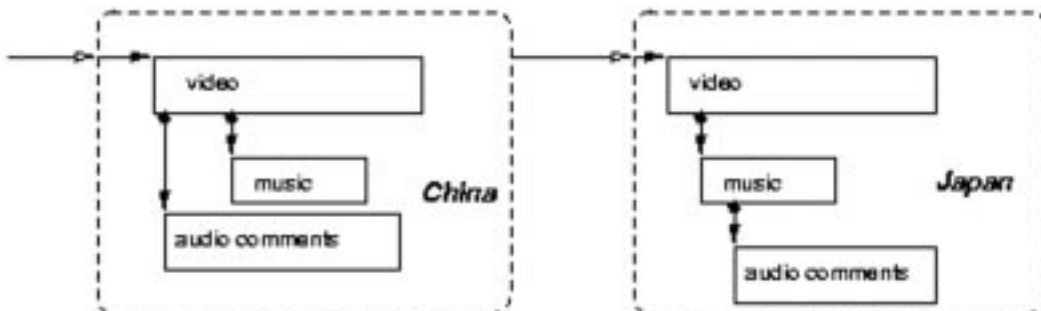

Fig. 2. Temporal composition of the example.

Example 1. Presentation of a travel agency.

```
<object id="travel-to-china">
      <object id="video" src="china-clip.mpeg">
            <time-point id="com" value="15" unit="second">
            <time-point id="mus" value="25" unit="second">
      </object>

      <object id="comments" src="china-comments.wav">
      </object>

      <object id="music" src="china-music.mid">
      </object>

      <htlink orig="beg" target="video.beg">
      <htlink orig="video.com" target="comments.beg">
      <htlink orig="video.mus" target="music.beg">
</object >

<object id="travel-to-japan">
      <object id="video" src="japan-clip.qt">
            <time-point id="mus" value="10" unit="second">
      </object>

      <object id="comments" src="japan-comments.au">
      </object>

      <object id="music" src="japan-music.mp3">
            <time-point id="com" value="30" unit="second">
      </object>

      <htlink orig="beg" target="video.beg">
      <htlink orig="video.mus" target="music.beg">
      <htlink orig="music.com" target="comments.beg">
</object>

<htlink target="travel-to-china.beg">
<htlink orig="travel-to-china.end" target="travel-to-japan.beg">
```

4. Support for user adaptable multimedia presentations

Our goal is to provide a generic solution for user adaptation of synchronized multimedia presentations. To adapt a generic presentation, the user specifies a predicate that, applied to the presentation, generates a customized view. So, we need two types of adaptation support: a way to express alternate content and content descriptions in the generic presentation, and a way for specifying user content predicates. We will present the adaptation support below.

The Resource Description Framework (RDF) [26] provides facilities related to our problem. RDF is a framework for specifying metadata that enables automated processing of World Wide Web resources. RDF uses XML as a common syntax for the exchange and processing of metadata. By exploiting the features of XML, RDF imposes structure that provides for the unambiguous expression of semantics and, as such, enables consistent encoding, exchange, and machine-processing of standardized metadata.

RDF provides the means of describing resources, their properties, and corresponding values, so that semantics can be associated with World Wide Web resources. Meaning in RDF is referred through a reference to a schema that specifies definitions and restrictions of usage. To avoid confusion between independent and possibly conflicting definitions of the same term, RDF uses the XML namespace facility to refer to various schemas.

Our adaptation support is partially inspired by the RDF; however, we need to add new elements and redefine some RDF elements. In this paper, we use a new namespace called AMP (Adaptable Multimedia Presentations):

```
<?xml namespace ns="http://www.imag.fr/AMP" prefix="amp">
```

In the rest of the paper, we assume that we use elements defined in the namespace and we skip the prefix.

4.1. Content descriptions

A generic multimedia presentation has to include descriptions of media objects or temporal links, so that the user may choose relevant components of the presentation. We use two forms of descriptions: short inline descriptions included as attributes in an element and RDF-like descriptions associated with an element through a reference.

Here is an example of an inline description:

```
<object id="china-video" src="china.mpg"
        description="a general presentation of China">
```

For very structured descriptions, we can associate a description using a reference:

```
<object id="china-video" href="#china-desc">
     ...
</object>

<description id="china-desc">
     <source src="china.mpg">
     <author>Bertolucci</author>
     <desc val="a general presentation of China">
     <duration val="60" unit="second">
     <format val="video/mpeg">
</description>
```

Note that in this example, we use a local reference to the description (`href="#china-desc"`), but we could have provided an URL to an external document as well (for example, a document generated from a database of descriptions).

Using descriptions, we can also associate semantics with temporal links:

```
<htlink target="travel-to-china.beg"
        description="long presentation">
```

4.2. Alternate content

In addition to content descriptions, we need a way to express alternate versions of an object or a hypertime link. A simple and common example of alternate content is the familiar **<img>** element:

```
<img src="an-image.gif" alt="W3C logo">
```

It is useful for providing alternate text for an image, but this way is insufficient for specifying two alternate versions of an image. SMIL provides the **<switch>** element for specifying alternate content; however, it cannot express two different versions of the temporal composition independently of media objects. In our case, we need to specify that one media object or hypertime link is an alternative version of another one. Using an element attribute is not enough, because we cannot define an alternate object in an inline attribute. Instead, we define alternate content in a similar way to content descriptions by using inline attributes to mark up different versions of the same object or by using an element similar to the RDF bag with alternate components:

```
description:
        < description id = "name" >
             alternatives
        < /description >


alternatives:
        < alt >
             alternative elements
        < /alt >


alternative elements:
        < li altval = " name "
             default >
             attributes
        < /li >


attributes:
        < source src = " url " >
        < importance val = " value " >
        < version val = " value " >
```

Here is an example of the inline declaration format:

```
<object id="china-video" version="long" importance="low"
        src="long-china-clip.mpeg">
     ...
</object>


<object id="china-video" version="short" importance="high"
        src="short-china-clip.mpeg">
     ...
</object>
```

The attribute **id** identifies an alternate object with different versions, **version** is a selector of a version, **importance** associates the information about its importance. The attributes can be used in a user condition to select an appropriate version. The following example illustrates how alternative hypertime links can be expressed:

```
<htlink id="china-link" version="long"
        orig="travel-to-china.end" target="travel-to-japan.beg ">
<htlink id="china-link" version="short"
        orig="travel-to-china.end" target="conclusions.beg">
```

Another way to express alternate content is to declare an object first, as shown in Section 4.1:

```
<object id="china-video" href="#china-desc">
     ...
</object>
```

and then specify alternate content in a description:

```
<description id="china-desc">
    <alt>
        <li altval="a0" default>
            <source src="long-china-clip.mpg">
            <importance val="high">
            <version val="long">
        </li>
        <li altval="a1">
            <source src="short-china-clip.mpg">
            <importance val="low">
            <version val="short">
        </li>
    </alt>
</description>
```

We have defined the **alt** element acting as a container with alternatives denoted by **li**. We have assigned distinctive values to each alternative by including an **altval** parameter and we have indicated a default alternative with the parameter **default**.

Content descriptors and alternatives can also be associated with layout objects which have the same characteristics as media objects.

4.3. Content predicates

The support for content descriptions and alternate content allows the author to annotate presentations so that they can be customized by the user. To do this, the user has to specify a predicate that can be matched against a generic presentation to yield a result document.

An emerging W3C activity on query languages is related to this problem. The activity deals with a more general problem of querying collections of HTML/XML documents described by means of RDF metadata [1,9,14]. Research done so far indicates that solutions based on query languages such as SQL may gain importance. In our case, we only need a way to express an assertion concerning content descriptors and alternate content. We propose to use a simplified RDF query similar to the SQL **where** clause for this goal:

```
condition:
        < condition >
            conjunction
            disjunction
        < /condition >

conjunction:
        < conjunction >
            selection
            choice
            conjunction
            disjunction
        < /conjunction >
disjunction:
        < disjunction >
            selection
            choice
            conjunction
            disjunction
        < /disjunction >

selection:
        < selection type  = " description-attribute "
                  value = " attribute-value " >

choice:
        < choice type  = " alt-attribute "
                value = " attribute-value " >
```

`<selection>` allows to select only the elements that match the value of the description attribute. Elements that do not match are not included in the result document. **`<choice>`** allows to choose one alternative out of several ones. Elements that are not expressed using alternatives are included by default in the result document.

The condition may include a conjunction or a disjunction of attribute values that should match the result elements. For example, we can have the following content predicate:

```
<condition>
    <conjunction>
        <choice type="version" value="short">
        <selection type="description" value="China">
    </conjunction>
</condition>
```

Although we have presented a simple way of specifying predicates, more powerful and sophisticated query languages [29] could be used for customizing generic presentations. A predicate is a high-level means for solving the adaptation problem, so the last question to answer is: who prepares and issues a predicate for customization? It can be done at least in three ways:
- *user profile* — a player automatically generates a predicate based on a user profile or checkboxed options (e.g. short or long version),

- *adaptation proxy* — a proxy that takes care of the characteristics of the client machine and its current network connectivity can issue a predicate adequate to the current conditions (e.g. a sequential version of a parallel presentation on a machine that does not have enough processing power),
- *advanced player interface* — a player can propose to the user a dialog menu to enter a condition (e.g. predicate on contents).

We suppose that the underlying communication protocol used for accessing a presentation will support transferring a predicate to the server. For example, we can use the POST method to transfer a predicate and obtain a customized presentation.

5. Examples

To illustrate the flexibility and expressing power of our support, we present below three examples of user conditions that yield a customized presentation (layouts are not specified for the sake of clarity). The conditions are applied to our motivating example presentation (Example 1) enriched with content descriptions and annotations for alternate content:

```
<object id="China" description="Travel offer in China">
    <object id="video" version="long" importance="low" default
            src="long-china-clip.mpeg">
        <time-point id="com" value="15" unit="second">
        <time-point id="mus" value="25" unit="second">
    </object >
    <object id="video" version="short" importance="high"
            src="short-china-clip.mpeg">
    </object>
    <object id="comments" src="china-comments.wav "> </object>
    <object id="music" src="china-music.mid"> </object>
    <htlink orig="beg" target="video.beg">
    <htlink version="long" importance="low" default
            orig="video.mus" target="music.beg">
    <htlink version="long" importance="low" default
            orig="music.com" target="comments.beg">
</object>

<object id="Japan" description="Travel offer in Japan">
    <object id="video" version="long" importance="low" default
            src="long-japan-clip.qt">
        <time-point id="mus" value="10" unit="second">
    </object>
    <object id="video" version="short" importance="high"
            src="short-japan-clip.qt">
    </object>
    <object id="comments" src="japan-comments.au"> </object>
    <object id="music" src="japan-music.mp3">
        <time-point id="com" value="30" unit="second">
    </object>
    <htlink orig="beg" target="video.beg">
```

```
        <htlink id="start-music" version="long" importance="low" default
              orig="video.mus" target="music.beg">
        <htlink id="start-comments" version="long" importance="low "default
              orig="music.com" target="comments.beg">
</object >

<htlink target="China.beg">
<htlink id="start-Japan" version="sequential" default orig="China.end"
        target="Japan.beg">
<htlink id="start-Japan" version="parallel" orig="China.beg" target="Japan.beg">
```

The following condition concerns only content descriptions and results in a presentation of journeys to China:

```
<condition>
      <selection type="description" value="China">
< /condition>
```

The result of the condition is the following:

```
<object id="China">
      <object id="video" src="long-china-clip.mpeg">
            <time-point id="com" value="15" unit="second">
            <time-point id="mus" value="25" unit="second">
      </object >
      <object id="comments" src="china-comments.wav"> </object>
      <object id="music" src="china-music.mid"> </object>
      <htlink orig="beg" target="video.beg" >
      <htlink orig="video.mus" target="music.beg">
      <htlink orig="music.com" target="comments.beg">
</object>

<htlink target="China.beg">
```

The following condition illustrates the use of alternate content and returns a short presentation of all journeys:

```
<condition>
      <choice type="version" value="short">
</condition>
```

The result document is the following:

```
<object id="China">
      <object id="video" src="short-china-clip.mpeg"> </object>
      <htlink orig="beg" target="video.beg">
</object >

<object id="Japan">
      <object id="video" src="short-japan-clip.qt"> </object>
      <htlink orig="beg" target="video.beg">
</object>
```

```
<htlink target="China.beg">
<htlink orig="China.end" target="Japan.beg">
```

The condition below shows how the temporal composition can be adapted:

```
<condition>
    <disjunction>
        <choice type="version" value="short">
        <choice type="version" value="parallel">
    </disjunction>
</condition>
```

The result is a document that presents the short versions of video on China and Japan in parallel:

```
<object id="China">
    <object id="video" src="short-china-clip.mpeg"> </object>
    <htlink orig="beg" target="video.beg">
</object>

<object id="Japan" >
    <object id="video" src="short-japan-clip.qt"> </object>
    <htlink orig="beg" target="video.beg">
</object>

<htlink target="China.beg">
<htlink orig="China.beg" target="Japan.beg">
```

6. Discussion and related work

The examples of the previous section show the flexibility and expressive power of our solution. Associating descriptors with content elements allows the author to indicate which elements can possibly be chosen from a generic presentation. Specifying alternate elements helps to create many possible interpretations of a given document. The support extends our temporal model in which media objects and temporal composition are clearly separated, so that content descriptions and alternatives can be integrated in a seamless way. Our approach avoids the need for explicit specification of all possible choices, which is the drawback of other propositions. We compare below our approach with other related work.

6.1. Adaptive hypermedia

A considerable amount of research has been devoted to the problem of *adaptive hypermedia*. Adap-tivity in hypermedia is proposed as a means to interact with users having different needs, background knowledge, interaction style, and cognitive characteristics [2,7,11,20]. It is also seen as a solution to information overflow and navigation through large information spaces and ordinary hypermedia. However, most of this work concerns structuring the information as a set of hypermedia documents that can be tailored to user needs. In our case, we want to generate different customized versions of a generic document.

6.2. SMIL

SMIL defines the `switch` element to specify a set of alternative elements [24]. The player evaluates the elements in the order in which they occur in the switch element; the first acceptable element is chosen. However, the intended use of the element is different than our goals: the choice of an alternative depends on a test attribute, such as *system-bitrate,*

system-captions, system-language, and some others. A multimedia presentation is not the right place to specify this kind of condition, because the conditions depend on the platform and the type of connectivity that a client can have with a content server. The number of possible combinations of attributes can be quite substantial; imagine 8 languages, 4 bitrates, 5 execution platforms, and you need to specify 160 possible combinations. As the playing conditions are known at playing time, they should be the subject of content negotiation between the playing client and a content server. The SMIL recommendation admits this point by stating that the switch element can be replaced by content negotiation with a server.

Moreover, SMIL mixes media object definitions with temporal composition, so that it is impossible to provide both alternatives on the content and the temporal composition of a presentation. When authoring using the switch element, we must either explicitly declare all possible choices or specify a composition (sequential or parallel) of switch alternatives as illustrated in the example below. In the example, we assume that we want to play two different alternatives for each country, China and Japan, and there is a test attribute (not specified in the example) that allows to choose one of the alternatives.

```
<switch>
    <seq>
        <video id="China-1">
        <video id="Japan-1">
    </seq>
    <seq>
        <video id="China-1">
        <video id="Japan-2">
    </seq>
    <seq>
        <video id="China-2">
        <video id="Japan-1">
    </seq>
    <seq>
        <video id="China-2">
        <video id="Japan-2">
    </seq>
</switch>
```

6.3. Channels

Bulterman has proposed the concept of *channels* to deal with user-centric adaptation of multimedia presentations [3]. A channel groups several independent media tracks that the user can turn on or off at will, thus providing some form of adaptation. The concept mimics TV broadcasts having sound tracks in different languages — the user can choose a desired sound track. Channels are useful; however, they are limited by the fact that media tracks should be independent and run in parallel.

6.4. Tellim

The Tellim system tries to adapt to the user by testing network connectivity and tracking user behaviour [8]. Based on a learning module and a database, it chooses a personalized elements of a presentation. A system like Tellim may benefit from our adaptation support, because it allows fine-grained representation of alternatives.

6.5. Other work

The problem of adaptation is related to the problem of creating video summaries [17]. A video summary should have a given duration and represent as well as possible the content of a video. However, determining what is the best choice of content depends on many factors — different applications or users may provide different selection criteria. For example, some users may be interested in all people appearing in a movie, others may want to see all locations or certain type of shots. Similarly to our goals, video summaries can be seen as an optimization problem with constraints: given a summary duration, find the components that maximize a value function (the function that measures the importance of video components).

In the context of multimedia databases, when retrieving results of a query, we might be interested in presenting them to the user as a synchronized presentation. In this case, the presentation can be created according to the user preferences or some other constraints [10]. Algebraic Video introduced the concept of a *virtual video node*, a sort of small functional program associated with a video shot [30].

An algebra of operators allows to specify temporal composition and add semantic descriptions to shots. Relevant shots can be chosen by querying a collection of nodes to form a new virtual video. MPEG-7 activity also concerns media descriptions and facilities for searching multimedia contents [12].

Much work has been done in the domain of format adaptation: how to adapt encoding of a given media type. Holtman et al. have defined extensions to HTTP to handle transparent content negotiation [6]. The extensions allow the client to specify which version of a given object should be delivered by a server. Similar functionality has been proposed by Kamada and Miyazaki [22]. Fox et al. have developed a distillation proxy that degrades several media types (images, video, postscript) to decrease the download time [5]. We have used an approach based on mobile agents to decrease the user perceived response time in a nomadic environment — low-bandwidth connectivity via GSM [4]. The problem of format adaptation arose early with multimedia conferences. When a conference is multicast on the MBONE, important performance benefits can be achieved with *scalable video*, the encoding based on the principle of successive refinement. The base information is sent first, then enhancement layers add more information to increase the quality of presentation. Destinations that are connected using different bitrates can choose the layers corresponding to their constraints. Two protocols are being developed for content negotiation of multimedia streams. Real-Time Streaming Protocol (RTSP) provides support for signalling and controlling media streams: acquiring the knowledge of streams available on a server and their characteristics, initiating a session, stopping and pausing a stream, terminating a session [19]. RTSP can be used for choosing a desired media-encoding to be used by the content server for delivery. Session Initiation Protocol (SIP) provides support for establishing and controlling multimedia conferences [18]. Content negotiation and format adaptation can always be used along with our solution to improve adaptability at the level of media objects.

7. Conclusion

We have proposed a generic solution for user adaptation of synchronized multimedia presenta-tions. We consider adaptation as a transformation problem: the user specifies a predicate that applied to a generic multimedia presentation yields a customized view of the presentation. We have specified a means of expressing content descriptions and alternate content in generic multimedia presentations. User adaptation is based on content predicates that a player uses to select alternate content or elements that match content descriptors. Several examples have shown flexibility and expressing power of the proposed approach.

Acknowledgements

References

[1] G. Arocena, A. Mendelzon and G. Mihaila, RDF Query Languages for the Web, W3C Query-Languages-Work-shop-19981203, http://www.w3.org/TandS/QL/QL98/pp.html

[2] P. Brusilovsky, Methods and techniques of adaptive hypermedia, Journal of User Modeling and User-Adaptive Interaction, http://umuai.informatik.uni-essen.de:80/home.html

[3] D.C.A. Bulterman, User-centered abstractions for adaptive hypermedia presentations, in: Proc. ACM Multimedia'98, pp. 247–256, Bristol, UK, 1998.

[4] X. Delord, S. Perret and A. Duda, Efficient mobile access to the WWW over GSM, in: Proc. ACM SIGOPS Workshop, Sintra, Portugal, September 1998.

[5] A. Fox and E.A. Brewer, Reducing WWW latency and bandwidth requirements by real-time distillation, in: Proc. 5th International WWW Conference, May 1996, Paris, France, http://www5conf.inria.fr/fich_html/slides/papers/PS16/P48/overview.htm

[6] K. Holtman and A. Mutz, Transparent Content Negotiation, in: IETF RFC 2295, ftp://ftp.isi.edu/in-notes/rfc2295.txt

[7] K. Höök, Å. Rudstrom and A. Waern, Edited adaptive hypermedia: combining human and machine intelligence to achieve filtered information, Flexible Hypertext Workshop (at ACM Hypertext'97).

[8] T. Joerding and K. Meissner, Intelligent multimedia presen-

tations in the Web: fun without annoyance, Research report, 1998, http://www.inf.tu.dresden.de/~tj4/reports/tellim1.html

[9] A. Malhotra and N. Sundaresan, RDF Query Specification, W3C Query-Language-Workshop-19981203, http://www.w3.org/TandS/QL/QL98/pp/rdfquery.html

[10] H. Martin, Specification of intentional presentations using an object-oriented database, in: Proc. Advanced Database Research and Development Series, Vol. 8, 1998, World Scientific.

[11] M.T. Maybury (Ed.), Intelligent Multimedia Information Retrieval, MIT Press, 1997.

[12] MRG (MPEG Requirements Group), MPEG-7: Context and Objectives (version 10 Atlantic City), ISO/IEC JTC1/SC29/WG11 N2460, http://drogo.cselt.it/mpeg/standards/mpeg-7/mpeg-7.htm

[13] P. Prescot, Introduction to DSSSL, Research report, 1997, http://itre.uwaterloo.ca:80/~papresco/dsssl/tutorial.html

[14] J. Robie, J. Lapp and D. Schach, XML Query Language (XQL), September 1998, http://www.w3.org/Style/XSL/Group/1998/09/XQL-proposal.html

[15] F. Rousseau and A. Duda, Synchronized multimedia for the WWW, in: Proc. WWW-7, Brisbane, Australia, April 15–18, 1998. (Also in: Computer Networks and ISDN Systems 30 (1998), 417–429.)

[16] F. Rousseau and A. Duda, An execution architecture for synchronized multimedia presentations, in: Proc. 3rd European Conference on Multimedia Applications, Services and Techniques, Berlin, May 1998.

[17] J. Saarela and B. Merialdo, Using content models to build audio–video summaries, Poster presentation and poster paper, in: Proc. ACM Multimedia'98, Bristol, September 1998.

[18] H. Schulzrinne, A comprehensive multimedia control architecture for the Internet, in: Proc. 7th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'97), St. Louis, MO, May 1997.

[19] H. Schulzrinne, Real Time Streaming Protocol (RTSP), Request for Comments (proposed standard) RFC 2326 Internet Engineering Task Force, April 1998.

[20] P.D. Stotts and R. Furruta, Dynamic adaptation of hypertext structure, in: Proc. 3rd ACM Conference on Hypertext, San Antonio, TX, December 1991.

[21] WAF (Wireless Applications Forum), WAP WAE Specification (Wireless Application Protocol, Wireless Application Environment Specification), April 1998, http://www.wapforum.org/

[22] W3C, Client-Specific Web Services by using User Agent Attributes, W3C NOTE-agent-attributes-19971230, http://www.w3.org/TR/NOTE-agent-attributes-971230.html

[23] W3C, Extensible Markup Language (XML) 1.0, W3C REC-xml-19980210, February 1998, http://www.w3.org/TR/1998/REC-xml-19980210.html

[24] W3C, Synchronized Multimedia Integration Language (SMIL) 1.0 Specification, W3C Recommendation, http://www.w3.org/TR/1998/REC-smil-19980615

[25] W3C, Document Object Model (HTML) Level 1, W3C REC-DOM-Level 1-19981001, http://www.w3.org/TR/REC-DOM-Level-1/Level-one.html

[26] W3C, Resource Description Framework (RDF), Model and Syntax Specification, W3C PR-rdf-syntax-19990105, http://www.w3.org/TR/1999/PR-rdf-syntax-19990105

[27] W3C, Web Accessibility Initiative (WAI), W3C WAI Working Group, http://www.w3.org/WAI/

[28] W3C, Cascading Style Sheets, Level 1 (CSS1), W3C Test-Level 1-19981002, http://www.w3.org/Style/CSS/Test/

[29] W3C, Query Languages Workshop, http://www.w3.org/TandS/QL/QL98/cfp.html

[30] R. Weiss, A. Duda and D.K. Gifford, Composition and search with a video algebra, IEEE Multimedia 2(1): 12–25, Spring 1995.

Franck Rousseau is a Postdoc Fellow at LSR–IMAG laboratory. Previously, he was a member of technical staff at the Open Group Research Institute in Grenoble. His current research interests include multimedia documents, architectures and communication.

J. Antonio García-Macías is a PhD candidate at LSR–IMAG laboratory in Grenoble. Previously, he was an industry consultant in telematics and associate professor in Computer Science. His current research interests include distributed multimedia systems, nomadic networking, adaptable hypermedia, and active networks.

José Valdeni de Lima is a Professor in the Computer Science Institute at the Federal University at Rio Grande do Sul, Brazil. Previously, he was a Visiting Professor at the LSR–IMAG laboratory in Grenoble. His current research interests include hypermedia systems, hyperdocuments, multimedia databases, and integration of workflow with documents management systems.

Andrzej Duda is a Professor at INPG (Institut National Polytechnique de Grenoble). He is a member of LSR–IMAG laboratory in Grenoble. Previously, he was a Visiting Scientist at the MIT Laboratory for Computer Science, a Research Scientist at CNRS, and an Assistant Professor at the Université de Paris-Sud. He worked on the design and performance evaluation of distributed systems and his current research interests include distributed multimedia systems, information access, resource discovery and new network applications.