Encoding Multimedia Presentation for User Preferences and Limited Environments

Tayeb Lemlouma, Nabil Layaïda

ENCODING MULTIMEDIA PRESENTATIONS FOR USER PREFERENCES AND LIMITED ENVIRONMENTS

Tayeb Lemlouma and *Nabil Layaïda*

OPERA Project, INRIA Rhône Alpes Research Unit
Tayeb.Lemlouma@inrialpes.fr Nabil.Layaida@inrialpes.fr

ABSTRACT

This paper discusses a new approach of generating TV-like multimedia presentations that are adapted to the target user preferences and to limited devices. Three main points are discussed: 1) The encoding of video presentations from a SMIL specification 2) The adaptation of the video content based on the user preferences, and 3) The delivery of adapted multimedia presentations. The used architecture includes a content server, an adaptation proxy and a set of small devices in the form of personal device assistants (PDA). These devices request the content through a wireless network. In order to show how the system behaves regarding the user preferences and capabilities, two negotiation dimensions are considered: the user language and the memory capability of the device. The first dimension is used to generate a content that can be understood by the target user, e.g. a video with subtitles written in the preferred language. The second dimension is chosen to solve the problem of the system blocking that usually happens when limited devices access rich multimedia presentations over the network.

1. INTRODUCTION

Several multimedia presentation models exist; they consider not only the media content but also other dimensions of the presentation: logical, spatial, temporal and hyperlink. The declarative specification of the multimedia presentations really represents an important advance in the multimedia handling field. This approach facilitates the manipulation and the processing of the presentation that has been considered as an atomic entity by classical approaches.

SMIL 2.0 has become a World Wide Web Consortium recommendation in 2001. It is the dominant representation in Web technology for describing timing and synchronization of multimedia presentations. A careful attention has been given, in the design of SMIL, to modularity and extensibility of the recommendation and three language profiles have been proposed [2]. In the context of multimedia content adaptation for user preferences, SMIL offers a set of interesting mechanisms that provide a better flexibility [7]. In this paper we exploit SMIL to ensure the encoding of the video content. The SMIL content control module [1] is used in order to generate a video presentation adapted to the user preferences. The paper addresses also the problem of video content adaptation for limited devices. The considered capability dimension is the memory of the target device. This dimension is chosen in order to avoid the blocking of limited users when they access to rich server multimedia presentations.

2. FROM SMIL TO VIDEO

The video generation from the SMIL specification includes the SMIL parsing and the video encoding.

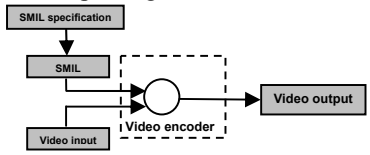

Figure 1: Video generation from the SMIL specification

The video encoding entity (Figure 1) includes the decoding of the original video to an uncompressed form (RGB 24 bit format), the application of our SMIL encoder and the generation of the output video. The used video decoder depends on which way the input video was encoded initially. The decoder is configured to generate an uncompressed form of each video frame which is simply the pixel representation of the video. Each pixel in the RBG form is represented by three bytes that correspond to the red, green and blue color value of the pixel. A line *l* of an original frame is represented by a set of *lineStride* byte values. Where *lineStride* = *pixelStride* x *videoWidth*. Here the *pixelStride* value equals to three (R, G and B components). Since a frame is a set of *videoHeight* lines, the size of an uncompressed frame equals to *lineStride* x *videoHeigh* bytes.

Since we discuss only TV-like multimedia presentations obtained after manipulating the original video; the SMIL encoder does not create pixels that are not covered by the video frame box. This means formally that an encoded pixel p (x, y) created by the SMIL encoder satisfies always the condition: $0 \leq x \leq videoWidth$ and $0 \leq y \leq videoHeight$. It is important to take into account this last assertion while authoring the SMIL

scenario, otherwise some parts of the SMIL objects will be ignored by the encoder. The role of the SMIL encoder is to determine: what are the video pixels to modify, how and when. The modification of video frames is dictated by the input SMIL specification of the aimed multimedia presentation. A video frame f is modified by the encoder if:

$$\exists\, o \in \text{Objects}\,(S), [\text{begin}\,(o), \text{end}\,(o)] \cap [\text{begin}\,(f), \text{end}\,(f)] \neq \varnothing \quad (1)$$

Where Objects (S) represents the set of the media objects included in the input SMIL specification. The set of the video pixels modified by the SMIL encoder at an instant t is given by:

$$\mathcal{P} = \{p(x, y),\ p \in o,\ o \in \text{Objects}\,(S), [\text{begin}\,(o), \text{end}\,(o)] \cap [\text{begin}\,(\text{frame}\,(t)), \text{end}\,(\text{frame}\,(t))] \neq \varnothing\} \quad (2)$$

Where frame (t) represents the video frame f identified by:

$$t \in [\text{begin}\,(f), \text{end}\,(f)]$$

The implementation of a SMIL encoder that follows the frames modification as it is described in Eq.1 can results, in some situations, to generate a video content that does not respect exactly the input SMIL timing. Indeed, following Eq.1 a media object can be displayed, on the final generated video, before the specified beginning instant or after the specified end instant. Example: Let us consider two video frames f_1 and f_2 where [begin (f_1), end (f_1)] = [10.08s, 10.12s] and [begin (f_2), end (f_2)] = [10.12s, 10.16s] and consider a SMIL specification that contains a media object o, such as [begin (o), end (o)] = [10.11s, 10.15s]. As we can note (Figure 2), both the intersections of the two frames intervals with the object interval are not empty. Consequently, the media object will be displayed 0.03s before the specified begin instant and 0.01s after the specified end instant.

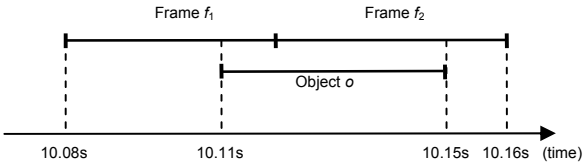

Figure 2: Example of interval intersections

To resolve this problem and in order to respect the SMIL timing, the encoder should create -when it is necessary- new video frames with the appropriate modifications. We distinguish three cases that require the creation of new frames. These cases are identified by the following conditions: A) begin (f) < begin (o) and end (f) ≤ end (o) B) begin (f) ≥ begin (o) and end (f) > end (o) C) begin (f) < begin (o) and end (f) > end (o). Where f represents the current frame processed by the SMIL encoder and o is a media object (it is possible to have more than one object) that satisfies: [begin (o), end (o)]∩[begin (f), end (f)] ≠ ∅.

The encoder behaves by applying the following algorithm: In the situation A, the original frame f is substituted by two new frames f_1 and f_2. The frame f_1 has exactly the same content as f, with [begin (f_1), end (f_1)] = [begin (f), begin (o)]. f_2 contains the content of f with the inclusion of the media object o, with [begin (f_2), end (f_2)] = [begin (o), end (f)]. In the situation B, f is substituted by f_1 and f_2 where [begin (f_1), end (f_1)] = [begin (f), end (o)] and [begin (f_2), end (f_2)] = [end (o), end (f)]. The content of f_2 is the same as f while the content of f_1 includes the content of o. The last situation, i.e. C, requires the substitution of the original frame f by three new frames f_1, f_2 and f_3. These frames are defined by: [begin (f_1), end (f_1)] = [begin (f), begin (o)], [begin (f_2), end (f_2)] = [begin (o), end (o)] and [begin (f_3), end (f_3)] = [end (o), end (f)]. The media object o is included within the frame f_2. The frames f_1 and f_3 have the same content as the original frame f.

The situation becomes more complicated, if the SMIL specification includes several media objects o_i that satisfy (when processing a frame f): [begin (o_i), end (o_i)]∩[begin (f), end (f)] ≠ ∅. The solution that we propose in this case is to consider each time only one media object and to apply the previous algorithm (applied in A, B and C situations) with the selected object. After applying the algorithm and eventually the creation of new frames, the concerned object is removed from the set of media objects and the algorithm is re-applied on another object from the set and so on until that the object set becomes empty. The order of media objects selection is not important. As we have seen, the SMIL encoder uses the media object content to respect the specified SMIL scenario. To avoid the encoding latency, the media objects are loaded to memory during the SMIL parsing (see Figure 1). Consequently, the effort done by the encoder is minimized which ensures a best quality of the video encoding especially when the encoding is done in a real time application.

3. SMIL ADAPTABILITY AND CONTENT NEGOTIATION

SMIL 2.0 Content Control Module [1] allows using an element called *switch* to specify inside the SMIL content a collection of alternative elements. The content selection can be expressed using the SMIL system test attributes (the seven attributes of SMIL 1.0 and the new five attributes added by SMIL 2.0). In our approach, we delegate the adaptation task to the proxy developed within the NAC architecture [3]. NAC is a proxy based architecture that enables the content adaptation and negotiation according to the users' capabilities and preferences. Here, the task of the proxy is limited to handle only SMIL and video content. The proxy includes two main modules: the player listener and the user context module (UCM) listener. The player listener is responsible to receive the user requests, to send them to original servers and to send the adapted answer (which can be the

same as the original answer of the server) to the user. The role of the UCM listener is to receive the profile and the profile changes of the users [4]. A profile contains a set of preferences and capabilities concerning the user [6], e.g. the screen size, the user agent, etc. This kind of information (called usually *negotiation dimensions*) is used to perform properly the task of content adaptation.

3.1. User preferences

The negotiation dimension - related to the user preferences - that we consider in this paper is represented by the system language of the user. Using the negotiation dimension communicated by the UCM module of the user, the proxy evaluates the SMIL switch element and chooses, when possible, the first acceptable element. An element (a timing structure or a media object) with no test attribute is always acceptable. Example: Let us consider the following part of a SMIL specification:

```
<switch>
<img region="subtitling_zone" src="subtitle1.gif" begin="14.5" end="15.9" systemLanguage="fr"/>
<img region="subtitling_zone" src="subtitle2.gif" begin="14.5" end="15.9" systemLanguage="en"/>
</switch>
```

Here, if the user prefers to receive the multimedia presentation in French, the proxy will adapt the previous part to the following:

```
<img region="subtitling_zone" src="subtitle1.gif" begin="14.5" end="15.9" />
```

The final SMIL presentation will so describe a scenario adapted to the target user. Thus from the same SMIL presentation, the proxy can generate different multimedia presentations: e.g. - with different TV logos or subtitles in different languages - by filtering some of the multimedia content based on the user type, etc. After the negotiation step, the SMIL specification is adapted to the user preferences. The proxy can send the new adapted multimedia presentation in the form of SMIL or video. The choice of the media type to send depends on the user agent of the target device (a video or a SMIL player).

3.2. User capabilities

The capability dimension that we consider is represented by the memory capability of the target device. This dimension is chosen in order to resolve the problem of the system blocking that happens usually when limited devices access rich multimedia presentations over the network. Indeed, embedded systems and user interfaces of these devices do not provide a good memory consumption control: applications still activated although they are not used, the UI does not make easy the consulting of the opened applications and documents, etc. Consequently, multimedia applications are usually subject of blocking when the available memory is not enough. Moreover, the user application has no idea about the scenario and the required resources of the multimedia presentation existing in the server side. One possible solution is the use of the proxy and making it responsible to transmit only multimedia data that can be played by the target device. To do so, we have enriched the UCM module by and additional functionality that allows reporting (on-demand and regularly) the memory capacity and the current memory state of the device. This helps the proxy: 1- to classify the device thanks to the whole capacity value of the memory, and 2- to follow the behavior and the current state of the user memory regarding the server multimedia application. The two following figures show an example of a SMIL presentation and the corresponding memory state of a pocketPC (iPAQ 3600) that access to the presentation using the HTTP protocol (through the NAC proxy) over a 802.11 wireless network. The used player is pocketSMIL[5].

```
<par>
  <video id="vid" region="region_video" src="Videos/orange4_f.mpg"/>
  <img region="img_zone1" src="img1.gif" begin="1s" end="vid.end"/>
  <img region="img_zone2" src="img2.gif" begin="2s" end="vid.end"/>
</par>
```

Figure 3: The server SMIL presentation

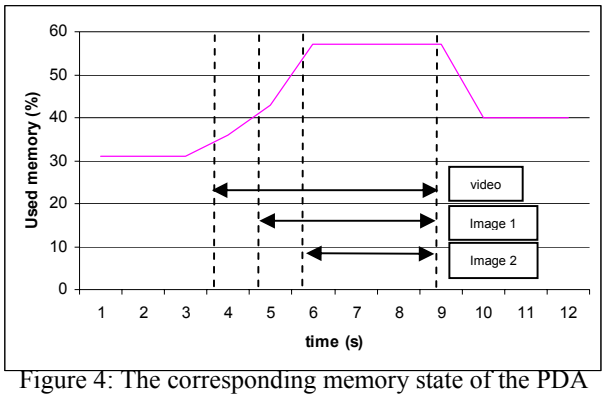

Figure 4: The corresponding memory state of the PDA (sent by the UCM module)

As we can note, the memory storage changes when the pocketSMIL starts to play the SMIL presentation. Naturally, the use of the memory increases when the player starts to download and decode the media objects. In this example the maximal used value represents 57% of the device memory. Actually, the memory-based adaptation of the proxy stills in the form of a prototype implementation. The considered adaptation is represented by a simple dropping of the video frames when the percentage of the used memory starts to be higher than a given value: α. This value depends to the device and can be estimated after several experimentations. It represents the limit of the memory size that should not be exceeded in order to avoid the device latency and blocking. For example, if α equals to 50% in the previous situation (Figure 4); the proxy drops the frames until the UCM reports a memory state that is less than α. When this happens, the new frame bit rate will be kept until receiving a memory state that allows increasing the current bit rate. This approach works fine and allows

avoiding the device blocking and the transmission of useless video frames. However, it presents two main problems: the estimation of the α value and the impossibility of decreasing continuously the video bit rate. The last problem shows that the adopted approach does not always guarantee the adaptation of the video. This happens generally when the device has a very small memory capacity or when the memory is overloaded independently to the multimedia application server.

4. EXPERIMENTAL RESULTS

In the following some experimental results concerning the generation of the video content from two SMIL specifications (the same scenario with different subtitles instances) and using two different input videos. Regarding the output video, the given measurements concern only the uncompressed generated video. The video stored in the server side is obtained by applying a compression method. The video compression is not the scope of this paper. The following table gives some characteristics of the two videos used in the SMIL representations.

Table1. The input videos

	Video 1	**Video 2**
Input format	MPEG 1	Indeo Video 3
Video dimensions	352x288	
Frame rate (fps)	25.0	
Video size (Kbytes)	8501	8770
Video duration (s)	30.99	30

The general form of the used SMIL specifications can be represented by the following temporal scenario:

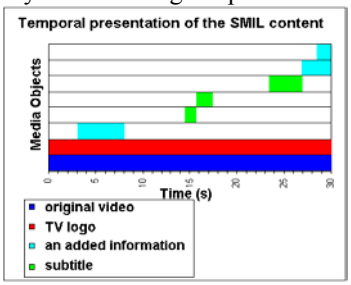

Figure 5: Temporal graph

Table2. The output videos

	Video 1	**Video 2**
Output format	uncompressed format (RGB 24-bit)	
Video dimensions	352x288	
Frame rate (fps)	25.8	25.0
Video size (Kbytes)	197918	227940
Video duration (s)	30.99	30
Frame Number	801	750
Frame duration (s)	0.0387	0.04
Encoding and storing time (ms)	35972	39577
Average encoding time (s/frame)	0.045	0.053

Table 2 shows the characteristics of the video content generated from the two used SMIL specifications. As we have seen in Section 2, the SMIL encoder uses the temporal scenario to generate the aimed video. Here are some screen shots from the generated video that respects the SMIL scenario.

(a) (b) (c) (d)

Figure 6: The encoded video

The proxy can generate adapted videos that respect the user preferences. Here is a screen shot from the video generated for a user that understands only English:

Figure 7: User preferences consideration

5. CONCLUSIONS

This paper combines the declarative definition of multimedia presentations with the video content encoding. A video encoding approach from a SMIL specification is presented. The SMIL adaptability and its declarative definition have allowed adapting automatically (using a proxy based architecture) the multimedia presentation to the user preferences. The generated presentation is in the form of a SMIL content or an encoded video. The same architecture was used to experiment a video adaptation for limited device in order to avoid the frequent devices blocking when they access to remote presentations.

6. REFERENCES

[1] Dick B. and Jeffrey A. SMIL 2.0 Content Control Modules. http://www.w3.org/TR/smil20/smil-content.html

[2] Layaïda N. and Van Ossenbruggen J. SMIL 2.0 Language Profile. http://www.w3.org/TR/smil20/smil20-profile.html.

[3] Lemlouma T. and Layaïda N. Adapted Content Delivery for Different Contexts. SAINT 2003 Conference, January 27-31 2003. Orlando, Florida, USA.

[4] Lemlouma T. and Layaïda N. Universal Profiling for Content Negotiation and Adaptation in Heterogeneous Environments. W3C Workshop on Delivery Context, W3C/INRIA Sophia-Antipolis, France, 4-5 March 2002.

[5] Pocket SMIL. http://opera.inrialpes.fr/pocketsmil/. INRIA Rhône Alpes. 2002.

[6] W3C. CC/PP. http://www.w3.org/TR/CCPP-struct-vocab/, W3C Working Draft 15 March 2001.

[7] W3C. SMIL 2.0 Basic Profile and Scalability Framework. http://www.w3.org/TR/smil20/smil-basic.html.