Adaptive Video and Metadata Display using Multimedia Documents

Cyril Concolato
Telecom ParisTech; Institut Telecom; CNRS LTCI
46, rue Barrault
75 634 Paris Cedex 13, France
cyril.concolato@telecom-paristech.fr

ABSTRACT
Adaptation of multimedia content is a problem which has been addressed from different points of view in existing works. The specific problem of adapting multimedia documents which describe the spatial, temporal and interactive organization of several media elements such as video, text, images and graphics is still an open problem. In this paper, we present a system capable of generating adaptive multimedia documents, based on templates, for the display of video content, annotated with information such as regions of interest, images and descriptive text. We present how we designed our templates, their structure and the adaptation logic, based on JavaScript. Some results are presented and analyzed.

Categories and Subject Descriptors
H.5.4 [**Information Systems**]: Hypertext/Hypermedia – *User issues* ; H.5.2 [**Information Systems**]: User Interfaces – *Graphical user interfaces (GUI), Screen design, User-centered design;* I.7.2 [**Computing Methodologies**]: Document Preparation – *Multi/mixed media, Scripting languages.*

General Terms
Algorithms, Design, Experimentation, Languages.

Keywords
Adaptation, Layout, Rich Media, Region of Interest, SVG, MPEG-4 BIFS.

1. INTRODUCTION
With the increasing number of sources and the diversity of devices to view multimedia content, personalization of such multimedia content, according to the user preferences and the usage environment is required to improve the user experience and to make browsing of multimedia content more efficient. Such personalization can be achieved using adaptation techniques. There exist several techniques to adapt elementary media elements such as video, audio, images or text for example using coding techniques (e.g. changing the format, the bitrate, using scalable codecs), using transmoding (changing the modality, from text to speech) or using bitstream switching (changing the language or the format of an audio track). However, when complex multimedia applications, that compose different elementary media together, need to be adapted, the adaptation problem is not so well addressed. Indeed, the adaptation of the composition of the media needs to take into account the nature of the elementary media; the spatial layout of the media and the viewing characteristics of the device (e.g. screen size); the temporal organization of the content; and finally, the interactive behavior of the content and the input characteristics of the device (keyboard, mouse …).

In this paper, we present our work regarding the adaptation of the composition of media elements when viewing a video and associated metadata on different devices, with different screens sizes. In our application, the main media elements to be shown are video streams. We assume that each video to be viewed has some metadata associated to it in the form of pairs of image and text. We show how the use of annotation of the video, and in particular regions of interest can be used to enable adequate viewing of both video and metadata. We also describe how a combination of existing multimedia document formats and the use of user interactions can enable the adapted viewing of the content.

The remainder of this paper is organized as follows. Section 2 presents related work in the area of multimedia document viewing and adaptation for similar applications to the one presented above. Section 3 describes the architecture of the content and the adaptation methodology. Section 4 presents some results and discusses them. Finally, Section 5 concludes this paper and presents future work.

2. RELATED WORKS
In the literature, we can find several types of works related to this paper. First, there are many publications related to the viewing of video on devices with small screen size. For example, in [1], Knoche et al. discuss what the best zooming factor and cut are for viewing video sequences of different types on a mobile phone. In this paper we consider that the best zooming and cut factor are the ones that allow the display of the region of interest selected by the user, without distortion of the aspect ratio. In [2], the same authors also discuss legibility issues when displaying text in video. In this paper, we will deal with text as a first class component of a multimedia document, to control its parameters such as font face and font size during the adaptation or personalization process.

In terms of adaptation of multimedia documents, several papers have also been published. For example, in [5], the authors discuss how declarative multimedia document can be used for dynamic

SAPMIA'10, October 29, 2010, Firenze, Italy.

zooming into annotated images. The work we present here is similar but uses video, instead of images, enriched with text and images and also it focuses more on the adaptation part than on the authoring part. In [3], the authors discuss how to author adaptive diagrams, where the adaptation algorithm is published using JavaScript. In this paper, our work reuses this concept but this paper adds the integration of different media elements, including video in the adaptation algorithm. Also, in previous work [6], we use the concept of scalable documents to achieve adaptation. Finally, we can also note that the MPEG-21 standard has addressed the problem of multimedia content adaptation through the Digital Item Adaptation standard. However, this work offers a granularity of adaptation that is not suitable for the adaptation of multimedia documents.

3. SYSTEM PRESENTATION

In this section we present the architecture of the system we use to generate adaptable rich media documents capable of showing video and metadata in an interactive manner, suitable for the user. This architecture is depicted in Figure 1. First, some offline (semi-automatic) annotation process (not shown in the figure) produces metadata associated with the video sequence. This metadata is composed of:

- The video file name, the width, the height, the duration and the frame rate of the sequence;
- A description of the detected regions of interest in the sequence. The regions of interest (ROI) can be faces, characters, or objects like cars, animals, etc. For each ROI a keyword is given. Additionally, ROI are typically moving in the sequence, so the description of the ROI is given for each frame in the video sequence by a position, a size;
- Finally, for each interesting object in the video, some textual content is provided. For each paragraph of text, an image is also associated.

The purpose of the generation tool is to produce a rich media document that is capable of:

- Displaying only the video in full-screen mode;
- Displaying the bounding boxes of each ROI, in a synchronous manner, on top of the video;
- Enabling the user to zoom on one of the ROI and to see the video sequence, with this view, without distorting the aspect ratio of the video;
- Displaying the video or a ROI together with the associated metadata;
- Adapting the layout of the metadata according to the device screen size.

In order to do that, we designed the system in the following way. First, a multimedia document enables the synchronized viewing of the ROI on top of the video stream. This first document points to the video stream and adds a graphical interactive stream on top of the video. This part of our work is already described in [6].

In this paper, we concentrate more on the second aspect, which is the adaptive display of metadata. For this part, we designed a second document, in this case using the Scalable Vector Graphics language. As indicated previously, we based the adaptation algorithm on principles similar to those expressed in [3]. We envisaged the layout of the metadata as a set of constraints and the solving of these constraints is performed on the client side, by some Javascript code, embedded in the document, which will modify the properties of the different media elements to produce the final layout, adapted to the screen. In our example application, we assumed that the author would give the following design rules:

- For each metadata pair, if both the text and the image can fit, the text shall be displayed at the right of the image;
- For each metadata pair, if both the text and the image cannot fit, the text is considered more important and the image shall be resized;
- For each image, if one of its resized dimension is smaller than 50 pixels, it shall not be displayed;
- If there is space for more than one metadata pair, the pairs should be displayed vertically;
- If all metadata pairs cannot be displayed, a 'Next' button should be used to interactively display the next pairs cyclically.

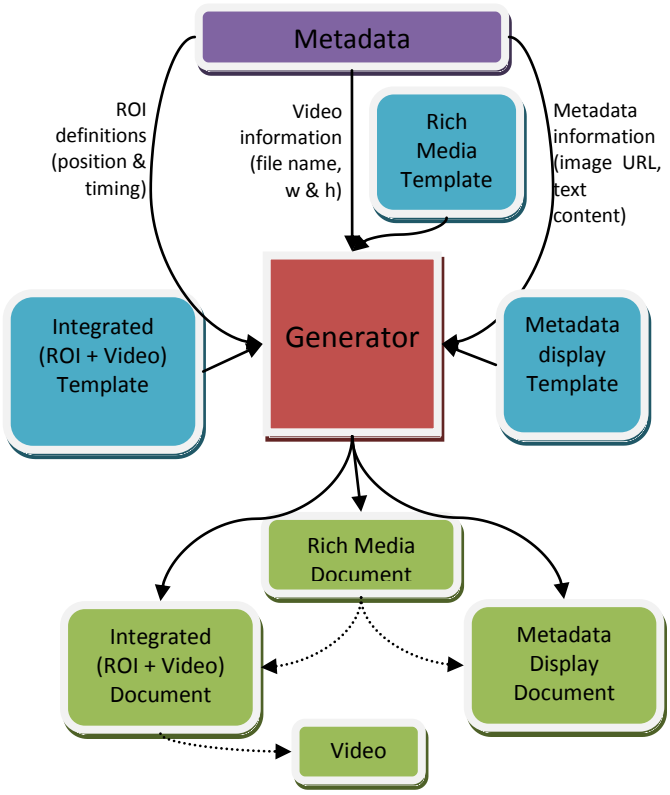

Figure 1 – Architecture for the generation of adaptable video and metadata scene

The adaptation of this metadata document is done by first capturing the resize event that happen either when the document is first loaded; or when the window is resized (e.g. on a PC); or when the document view is resized (e.g. when document is embedded in another one as described below). Given the display size of the document, we need to compute the size of each paragraph, when rendered in different rectangle (depending on

whether the accompanying image is present and depending on its size). For that we use the heuristic presented in [4].

Finally, both documents, the one for the display of the ROI and the one for the display of the metadata are aggregated in a single document that enables interactively to switch between the two viewing modes: video and ROI only, or video and ROI and metadata. In that case, again some design rules are assumed:

- When displaying metadata, the area reserved for the metadata should represent $1/4^{th}$ (resp. $1/3^{rd}$) of the screen, if the screen aspect ratio is bigger than one (resp. smaller);
- If the aspect ratio of the video (or the ROI) is larger than one, then the integrated video and ROI document should be displayed on the left and the metadata document on the right;
- Otherwise, the integrated video and ROI document should be displayed on top and the metadata document at the bottom.

For this global document, we designed it also using embedded Javascript code, first to detect user events and trigger viewing mode; then to detect resize events and propagate it to each embedded document.

4. IMPLEMENTATION, RESULTS AND DISCUSSIONS

4.1 Implementation

We implemented the system presented in Section 3 with a set of tools. The first tool is a C program that reads the input metadata (in XML), reads a template, generates the integrated Video+ROI document, and then encodes this document in the form an MPEG-4 BIFS document and packages the associated video in an MP4 file. The encoding and packaging steps are not mandatory but they enable a more efficient reading and insure a better synchronization.

The second tool is an XSLT transformation which generates a Scalable Vector Graphics document based on the metadata input document. This resulting SVG document points to a JavaScript file common for all video sequences that contains the adaptation logic.

Note that the general document integrating both sub documents does not need to be regenerated, since it does not depend on the metadata. This document was also described using the MPEG-4 BIFS language.

4.2 Results

Figure 2 shows how the display of the global document looks like when viewed in the GPAC player [7], in the viewing mode where only the video and the ROI are displayed. A light-bulb-shaped icon is a button that allows switching back and forth between this viewing mode and the metadata viewing mode. The other icon allows going back to a main menu to select another video.

Figure 3 shows the other viewing mode, i.e. video and metadata, when the aspect ratio of the screen is very tall. Notice that the video is displayed at the top, and the metadata at the bottom. You can also see that the metadata text is displayed at the right side of the image. You can see a limitation of the rules given in previous section, namely that the text could be flowed below the image for a better use of the space. You can also note the 'Next' button, which allows showing the other metadata (text and image) pairs.

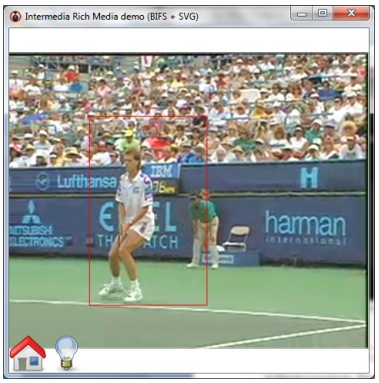

Figure 2 – Synchronized display of a moving region of interest on top of a video

Figure 3 – Video and metadata display on a tall screen

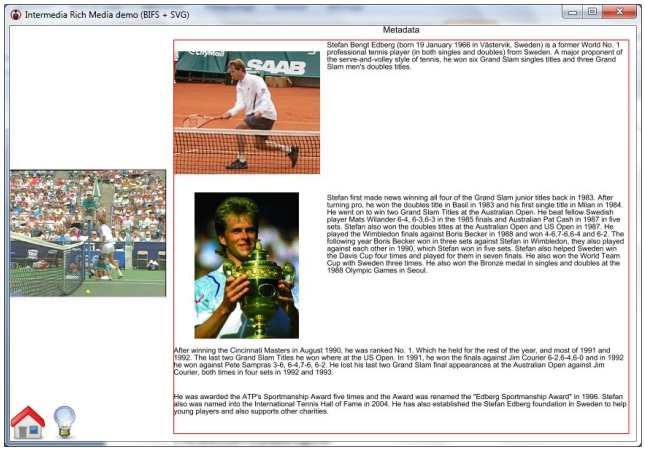

Figure 4 – Video and metadata display on a large wide screen

Figure 4 shows a different screen configuration. On this large wide screen, all metadata information is displayed (the 'Next' button is not present), but there is not enough space to display all images with a sufficient size, according to the given author rules, so some images are not displayed.

Figure 5 – Simultaneous display of a moving ROI and of associated metadata

Figure 5 shows a montage where three different snapshots of the player are displayed together. This figure highlights the fact the display area of the video and of the metadata are fixed according to our rules (in this case $1/4^{th}$), and that the display of the ROI is adaptive to the size of ROI and that the ROI aspect is preserved.

4.3 Discussions

The above results show that our system is functional. However, some improvements could be made. First, to some extent, the design rules that we assumed are too simple and lead to a rudimentary layout that is not perfect in all screen configuration. Then, taking the viewing distance, or some advance screen properties like dpi (dot-per-inch), could be used to determine the best image and font size for the user. In terms of ROI display, the interface is again simple and should be modified to support multiple ROI or ROI that appear and disappear too quickly. Finally, we should work on a merging of the two separate documents (one for the video and one for the metadata) to facilitate the display of timed metadata, synchronized with the video or with the selected ROI.

We believe that the way we implemented this system and designed the different documents is interesting because it actually combines different multimedia language, taking the best of each of them. For example, the MPEG-4 BIFS language is used for its capacity to describe frame-based synchronization, in a streaming manner. The SVG language was used for its handling of text and its support for JavaScript.

5. CONCLUSION

This paper has illustrated how adaptation of multimedia documents, composed of multiple media elements of different types, can be achieved, taking into account the results of an annotation process and also taking into account the user wishes, by means of interactions. This shows that adaptation of multimedia documents can improve the user experience but it also shows the need for a proper authoring tool where an author would be able to control the different adapted rendering, possibly to adjust the adaptation rules. The general problem of authoring adaptable content is a challenge that we want to address in future work.

6. ACKNOWLEDGMENTS

The author would like to thank the InterMedia partners for their help in setting the whole system in place. The work described in this paper was cofounded by the European Union (within the framework of the NoE INTERMEDIA, IST-038419).

7. REFERENCES

[1] Knoche, H. and Sasse, M. A. 2009. The big picture on small screens delivering acceptable video quality in mobile TV. *ACM Trans. Multimedia Comput. Commun. Appl.* 5, 3 (Aug. 2009), 1-27. DOI= http://doi.acm.org/10.1145/1556134.1556137

[2] Knoche, H. O., McCarthy, J. D., and Sasse, M. A. 2006. Reading the fine print: the effect of text legibility on perceived video quality in mobile tv. In *Proceedings of the 14th Annual ACM international Conference on Multimedia* (Santa Barbara, CA, USA, October 23 - 27, 2006). MULTIMEDIA '06. ACM, New York, NY, 727-730. DOI= http://doi.acm.org/10.1145/1180639.1180794

[3] McCormack, C., Marriott, K., and Meyer, B. 2008. Authoring adaptive diagrams. In Proceeding of the Eighth ACM Symposium on Document Engineering (Sao Paulo, Brazil, September 16 - 19, 2008). DocEng '08. ACM, New York, NY, 154-163. DOI= http://doi.acm.org/10.1145/1410140.1410172

[4] Hurst, N. and Marriott, K. 2007. Approximating text by its area. In Proceedings of the 2007 ACM Symposium on Document Engineering (Winnipeg, Manitoba, Canada, August 28 - 31, 2007). DocEng '07. ACM, New York, NY, 147-150. DOI= http://doi.acm.org/10.1145/1284420.1284458

[5] Kuijk, F., Guimarães, R. L., Cesar, P., and Bulterman, D. C. 2009. Adding dynamic visual manipulations to declarative multimedia documents. In Proceedings of the 9th ACM Symposium on Document Engineering (Munich, Germany, September 16 - 18, 2009). DocEng '09. ACM, New York, NY, 149-152. DOI= http://doi.acm.org/10.1145/1600193.1600227

[6] De Bruyne, S., Hosten, P., Concolato, C., Asbach, M., De Cock, J.; Unger, M., Le Feuvre, J., Van de Walle, R. Annotation based personalized adaptation and presentation of videos for mobile applications. *Mutlimedia Tools and Applications.* To appear.

[7] Le Feuvre, J., Concolato, C., and Moissinac, J. 2007. GPAC: open source multimedia framework. In Proceedings of the 15th international Conference on Multimedia (Augsburg, Germany, September 25 - 29, 2007). MULTIMEDIA '07. ACM, New York, NY, 1009-1012. DOI= http://doi.acm.org/10.1145/1291233.1291452