Proceedings of IEEE International Conference on Image Processing, ICIP'03, vol. 3, pp. 45-48, Barcelona, Spain, September 14-17, 2003.

MPEG–BASED PERSONALIZED CONTENT DELIVERY

Olivier Steiger, Touradj Ebrahimi

Swiss Federal Institute of Technology (EPFL)
CH-1015 Lausanne, Switzerland
{olivier.steiger, touradj.ebrahimi}@epfl.ch

David Marimón Sanjuán

Universitat Politècnica de Catalunya
Telecommunication Engineering School
E-08034 Barcelona, Spain

ABSTRACT

In this paper, we present a personalized multimedia content delivery system dealing with both user preferences and terminal/network capabilities. In order to ease interoperability with third-party applications, content annotation, user preferences handling and terminal/network capabilities description are managed by MPEG–7 and MPEG–21 standards. Offline content adaptation and annotation tools are proposed for content preparation. Content delivery is handled by a client-server architecture. This architecture is built around custom personalization and adaptation tools, and a commercial content streamer. The proposed system has been used to provide Universal Multimedia Access within the EC–funded R&D project PERSEO.

1. INTRODUCTION

The support of personalization has become one of the major success factors in current internet services. Personalization means to customize the functionality of the service based on the specific requirements and needs of individual users, channel capacity and terminal capabilities, in order to filter or adapt multimedia content. In application domains like TV-on-demand, the WWW and mobile telecommunications, personalization enables different users and terminals to access rich multimedia content transparently. This is referred to as Universal Multimedia Access (UMA) [1].

This paper describes a personalized content delivery system that is based on ISO's Moving Picture Experts Group (MPEG) standards family. The realization of tools for both content preparation and delivery is presented in Section 2 and discussed in Section 3. The research areas that are being addressed notably include content analysis, multimedia databases and dynamic architectures. The use of MPEG standards for content coding, annotation and management is preferable as it facilitates data exchange with third-party applications and benefits from the MPEG experience.

The concepts underlying the present work were set out by Perkis *et al.* in a paper on UMA from wired and wireless systems [1]. In their paper [2], Fossbakk *et al.* describe a test bed for simulating a UMA enabled system using members of the MPEG family for its realization. This work studies the integration of MPEG–4 and MPEG–7 within the MPEG–21 Multimedia Framework to adapt content to terminal and network capabilities. It does not however consider user preferences. A similar system which does not rely on standards is described in [3]. Content adaptation to different user profiles for digest video delivery [4] and mobile multimedia services [5] has been reported on as well. The integration of tools for both user preferences *and* terminal/network capabilities handling within a unique standards-based framework is original.

2. PERSONALIZED CONTENT DELIVERY SYSTEM

This Section proposes an architecture for personalized content delivery handling both terminal and network capabilities, and user preference. Content annotation, user preferences handling and terminal/network capabilities description are managed by MPEG standards in order to ease interoperability with third-party applications. MPEG standards have also been tested in numerous experiments and applications, and refined accordingly. The proposed tools are subdivided into two complementary categories: *content preparation* and *content delivery* tools. The former are used to generate *variations*, that is to say different resolution, format and modality versions, of the original content, and to annotate content using MPEG–7 and MPEG–21. The latter handles content delivery itself. All tools are schematically depicted in Figure 1. Their realization is detailed in Sections 2.1 and 2.2.

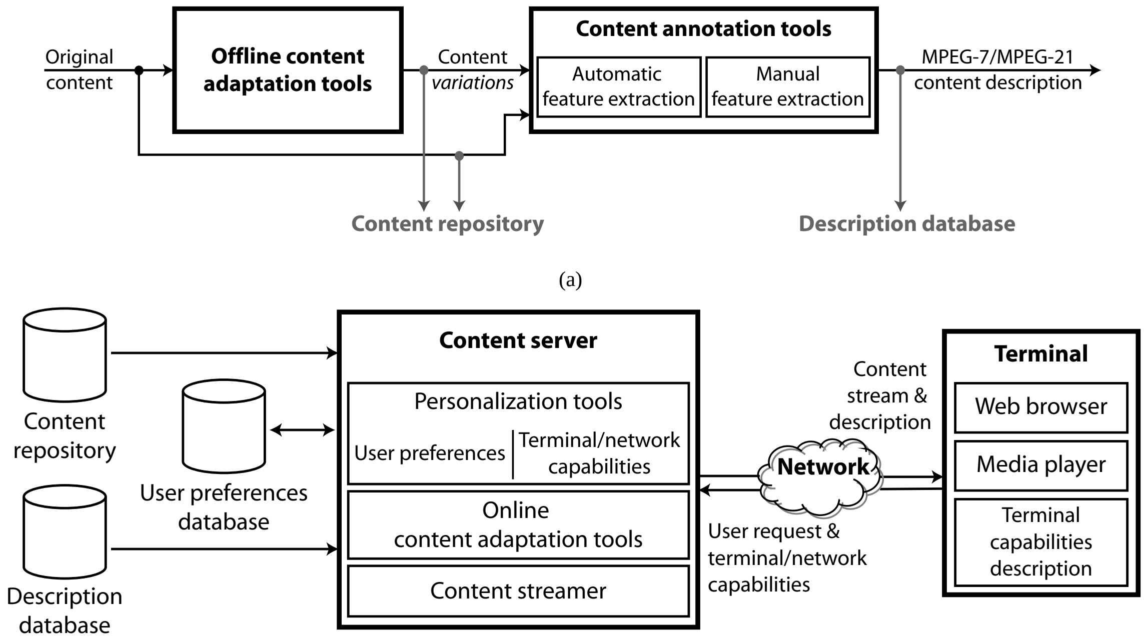
2.1. Content Preparation

Multimedia content is usually available in a single modality, resolution and format. To enable ubiquitous access, content variations must be generated either prior to delivery or on-the-fly. Content must also be annotated for browsing and retrieval. Tools for both offline content adaptation and annotation are proposed next.

2.1.1. Offline content adaptation

The goal of content adaptation tools is to adapt the original modality, resolution and format of multimedia content to potential terminal and network capabilities. The so-generated *variations* are in turn stored in the content repository. This progressive data representation scheme is called the *InfoPyramid*. Unlike online adaptation, offline adaptation is not done in real-time, but prior to streaming. Offline adaptation tools must be able to convert a large number of parameters, notably target bitrate, coding algorithm, audio sampling rate, image resolution and frame rate. In our system, we use Adobe *Premiere* and *Photoshop* (`http://www.adobe.com`) for audio/video and image adaptations, respectively. These products have been selected because they offer rich functionalities. Any other equivalent products may be used instead.

This work was supported in part by the European Commission under contract IST-2000-28443 (PERSEO).



(a)

(b)

Fig. 1. Personalized content delivery system. (a) Content preparation tools; (b) Content delivery tools.

2.1.2. *Content annotation*

Efficient multimedia browsing and retrieval presupposes the availability of content *metadata* (data about data). The MPEG–7 *Multimedia Content Description Interface* specifies a standard set of *Descriptors* (D) and *Description Schemes* (DS) to describe a variety of multimedia information [6]. As handwriting MPEG–7 descriptions is a laborious process, user-friendly content annotation tools are needed to generate them. In Table 1, an MPEG–7 descriptor set targeting personalized content delivery is proposed. The set includes descriptors for technical (media information, structure description, access tools) and content-related information (creation information, usage description, semantic description). The annotation tools automatically extract certain features using metadata contained in the content file header (e.g., file format, resolution, ...) and unsupervised analysis methods. Remaining features are completed by manual processing through a user-friendly Graphical User Interface (GUI). At the end, all content variations (Section 2.1.1) are grouped in an MPEG–21 *Digital Item* [7].

Although a few MPEG–7 annotation tools are publicly available, notably by IBM, we have implemented our own tools in order to get the necessary functionalities. For the sake of portability, the realization has been done in Java (`http://java.sun.com`). The GUI was built using Java Swing, and MPEG annotation is handled by the Java API for XML processing JAXP. Apple's QuickTime (`http://quicktime.apple.com`) media player is used for content playback because of the wide variety of readable formats. The *QuickTime for Java* API permits the integration of the player in the annotation software. Video keyframes and scenes are extracted by an automatic summarization algorithm [8].

2.2. Content Delivery

In a typical personalized content delivery scenario, the following actions are taken:

1. An end user registers to the system using a unique ID and password,
2. content is ordered according to the user preferences,
3. best content variations are selected according to terminal/network capabilities,
4. the sorted and filtered content list is sent to the user,
5. the user may optionally query content using refined search criteria,
6. content selected by the user is streamed,
7. user preferences are automatically updated according to the selection.

The tools implemented to realize this scenario on the server (Section 2.2.1) and client sides (Section 2.2.2) are set out next.

2.2.1. *Server*

Content, content descriptions and user preferences are stored in three distinct **databases** that are accessed by the content server. The *content repository* takes the original content and its variations. Since no querying is performed on the content itself, a file repository is sufficient. The *description database* stores the MPEG content descriptions. This database is queried by the personalization tools to find the optimal content-client match, therefore efficient sorting and filtering are required. In our system, we use the open-source XML database Xindice (`http://xml.apache.org/xindice/`). This product specifically targets large collections of XML files (like Digital Items) and uses W3C XPath for its query language, making it easily interchangeable. Since user preferences

DESCRIPTOR	PURPOSE	EXTRACTION
Description metadata		
`DescriptionMetadata`	Description version, author and history.	Manual
Media information		
`MediaIdentification D`	Uniquely identifies content.	Manual
`MediaFormat D`	Information about storage medium, media file and content coding.	Automatic
`MediaQuality D`	Perceptual quality rating.	Automatic
`MediaInstance DS`	URL of content file.	Automatic
Creation information		
`Creation DS`	Content title, abstract, creator, creation coordinates and creation tool.	Manual
`Classification DS`	Content form, genre, subject, purpose, language, release, audience and review.	Manual
Usage description		
`Rights datatype`	Information about right holders and content usage.	Manual
Semantic description		
`AgentObject DS`	*Who*: describes objects that are persons or organizations in a narrative world.	Manual
`Event DS`	*What*: describes a semantic activity in a narrative world.	Manual
`SemanticPlace DS`	*Where*: describes location in a narrative world.	Manual
`SemanticTime DS`	*When*: describes time in a narrative world.	Manual
Structure description		
`Segment DS`	Specifies temporal segments in multimedia content.	Automatic
Access tools		
`SummarySegmentGroup DS`	Defines video summaries using key frames and key sounds.	Automatic

Table 1. MPEG–7 descriptors for content annotation targeting personalized delivery.

and the usage history are described using MPEG–7 *user interaction tools* [6], the *user preferences database* has similar requirements. Therefore, Xindice is used here as well.

The **content server** groups tools for personalization, online content adaptation and content streaming. The *personalization tools* fulfill three distinct functions: handle user preferences, find the variations that best fit the present teminal/network capabilities, and generate personalized web pages for content browsing. User preferences handling involves two operations: user preferences acquisition and automatic update. Initial preferences are acquired by means of of a form. This form asks the user for his preference in categories like content topic, language, source, . . . The user must also give a priority to each preference. His answers are then translated into MPEG–7 *user preferences* and stored in the corresponding database. When consented to by the user, user actions (i.e., list of accessed content) are collected and stored in an MPEG–7 *usage history*. This data is used to update user preferences automatically by giving higher priority to content categories that are often selected, or by adding them to the initial preferences. The content is then ordered by matching the user preferences with the content-related information given by the MPEG–7 annotation (Section 2.1.2). Content variations that best fit the present usage environment are found by comparing their `MediaFormat` (Table 1) to the current terminal/network capabilities. This is done according to a preestablished set of rules[1]. To generate personalized web pages, terminal-specific XSLT transformations are applied to XML pages. In our system, the web pages are adapted using Apache Cocoon (`http://xml.apache.org/cocoon/`) and delivered by Apache Tomcat (`http://jakarta.apache.org/tomcat/`). *Online content adaptation* is the real-time adaptation of content to the usage environment just before streaming. This is mainly doable for format transcoding, while more complex adaptations must be done offline (Section 2.1.1). In our realization, Windows Media Encoder (`http://www.microsoft.com/windowsmedia/`) and Real Helix Producer SDK (`http://www.realnetworks.com/`) transform various content formats into Windows Media or Real streams. The *content streamer* finally sends the selected content to the user during playback. The streamer should support a wide variety of audio, image and video formats in order to feed different terminals, and must handle varying network conditions. For maximum flexibility, all Real Helix Server, Java Media Framework, Windows Media Server or QuickTime Server can be called depending on the needs in our realization.

[1]For instance, *if* content resolution > terminal resolution *then* content is not playable.

2.2.2. Client

A wide variety of clients is expected to access the system. Possible terminals include Personal Computers (PC), Personal Digital Assistants (PDA) and cell phones, connected over both wired (LAN, PSTN) and wireless networks (WLAN, GPRS/UMTS). Some of these devices have very limited resources, so client requirements should be as nonrestrictive as possible. On the **terminal**, a standard web browser runs the user interface. Content playback is done using any media player. Finally, some memory space has to be allocated to store the terminal capabilities description. Terminal capabilities are described using the MPEG–21 DIA *usage environment description tools* [9]. This data is accessed by the server to determine the most appropriate variations to be streamed. The **network** can be of any type as long as streaming is practicable. Dynamic network conditions (e.g., current bitrate) are determined by the server on-the-fly at streaming, while static conditions (e.g., maximum bitrate) are also described by MPEG–21 DIA. In our system, a PC, a color PDA and a B&W cell phone are simulated on a standard PC running Internet Explorer 6 and Windows Media

Player. Specific terminal capability files are stored for each simulated device. LAN, WLAN and PSTN networks are being used for connection with the server.

3. DISCUSSION AND CONCLUSIONS

The system proposed in Section 2 provides fully operational personalized content delivery. It is a prototype of the Universal Multimedia Access services that will be deployed by the EC-funded R&D project PERSEO. These services will use advanced user preferences handling schemes. Actual GPRS mobile terminals will be used, and all content preparation tools are to be integrated within a unique GUI. To test the proposed system, a content set with 2 long video sequences, 10 short video sequences, 10 still images and 5 audio titles is used (Table 2). A *high quality*, a *medium quality* and a *low quality* variation of each content were generated using offline adaptation tools (Section 2.1.1). All variations have been properly annotated using MPEG–7 (Section 2.1.2).

NAME	DESCRIPTION	AMOUNT
Long videos	News program	2
Short videos	Sports, ad, trailers	10
`High quality`	Real, 768x576, 25fps, color, 96kbps audio	
`Medium quality`	Real, 352x288, 15fps, color, 32kbps audio	
`Low quality`	Real, 176x144, 10fps, B&W, 11kbps audio	
Still images	Portraits, buildings	10
`High quality`	JPEG, 1024x768, color	
`Medium quality`	JPEG, 640x480, color	
`Low quality`	JPEG, 176x144, B&W	
Audio	Music	5
`High quality`	MP3, 192kbps	
`Medium quality`	MP3, 64kbps	
`Low quality`	MP3, 32kbps	

Table 2. Test content set.

We access the content over LAN, WLAN and PSTN networks using a simulated PC, a color PDA and a B&W cell phone (Section 2.2.2). Three different user profiles were created. The response of the system to the different terminals is satisfactory: the high quality variation is streamed to the PC, while the PDA gets medium quality and the cell phone low quality. If no adequate variation is available for a given terminal, the content would not be listed. For video and audio, all terminals receive the low quality variations over the PSTN network. A more refined response to network conditions is obtained by taking advantage of the multiple-stream feature of some formats (e.g., Real, Windows Media). The streaming server then selects the most appropriate stream automatically in the course of streaming. Content sorting according to user preferences is satisfactory as well. Content belonging to the user's favorite categories is listed first, followed by preferred categories with lower priority. Content that does not belong to any preferred category is not sorted and listed at the end. Despite the simplicity of the current implementation, automatic preferences update works fine as well. Content categories that are often selected are added to the user preferences or gain higher priority.

Some drawbacks of the system are highlighted by the tests. Due to the usage of an XML database, content filtering and sorting become slow on large data sets. This could be sped up by tabulating the description data. More powerful online content adaptation tools, such as real-time resolution and frame rate adaptation or automatic modality conversion, would diminish required storage space and content preparation time.

In conclusion, the presented system combines different tools and recent standards to provide personalized content delivery. This makes it an essential constituent of upcoming news distribution, TV-on-demand and mobile multimedia applications.

4. ACKNOWLEDGEMENTS

This work has been funded in part by the European Commission under contract IST-2000-28443 – Personalized multichannel services for advanced multimedia stream management (`http://www.perseoproject.org`).

5. REFERENCES

[1] A. Perkis, Y. Abdeljaoued, C. Christopoulos, T. Ebrahimi, and J.F. Chicharo, "Universal multimedia access from wired and wireless systems," *Birkhauser Boston trans. on Circuits, Systems and Signal Processing*, vol. 10, no. 3, pp. 387–402, 2001.

[2] E. Fossbakk, P. Manzanares, J.L. Yago, and A. Perkis, "An mpeg–21 framework for streaming media," *IEEE Fourth Workshop on Multimedia Signal Processing*, pp. 147–152, 2001.

[3] R. Mohan, J.R. Smith, and C.-S. Li, "Adapting multimedia internet content for universal access," *IEEE Trans. on Multimedia*, vol. 1, no. 1, pp. 104–114, 1999.

[4] T. Echigo, K. Masumitsu, M. Teraguchi, M. Etoh, and S.-I. Sekiguchi, "Personalized delivery of digest video managed on mpeg–7," *Proc. Int. Conf. on Information Technology: Coding and Computing*, pp. 216–220, 2001.

[5] Y.-F. Chen, H. Huang, R. Jana, S. John, S. Jora, A. Reibman, and B. Wei, "Personalized multimedia services using a mobile service platform," *Wireless Communications and Networking Conference WCNC2002*, vol. 2, pp. 918–925, 2002.

[6] P. van Beek, A.B. Benitez, J. Heuer, J. Martinez, P. Salembier, Y. Shibata, J.R. Smith, and T. Walker, "Text of 15938-5/fdis information technology – multimedia content description interface – part 5: Multimedia description schemes," Tech. Rep. N4242, ISO/IEC JTC1/SC29/WG11, Sydney, AU, July 2001.

[7] V. Iverson, T. Schwartz, Y.-W. Song, R. van de Walle, E. Santos, and D. Chang, "Text of 21000-2/fdis information technology – multimedia framework – part 2: Digital item declaration," Tech. Rep. N4813, ISO/IEC JTC1/SC29/WG11, Fairfax, VA, USA, May 2002.

[8] Y. Abdeljaoued, T. Ebrahimi, C. Christopoulos, and I.M. Ivars, "Video summarization for universal multimedia access applications," Tech. Rep. M5105, ISO/IEC JTC1/SC29/WG11, Melbourne, AU, October 1999.

[9] A. Vetro, A. Perkis, and S. Devillers, "Text of 21000-7/cd information technology – multimedia framework – part 7: Digital item adaptation," Tech. Rep. N5353, ISO/IEC JTC1/SC29/WG11, Awaji Island, JP, December 2002.