A QOS-DRIVEN UMA VIEWER IN MPEG-21 FRAMEWORK

Andrew Perkis *R. Venkatesh Babu* *Peder Drege*

Centre for Quantifiable Quality of Service in Communication Systems
Norwegian University of Science and Technology
Trondheim, Norway
e-mail: andrew@tele.ntnu.no, venkat@Q2S.ntnu.no, drege@stud.ntnu.no

ABSTRACT

This paper[1] presents a design and streaming architecture model of a Universal Multimedia Access (UMA) viewer[2] capable of handling multimedia presentations in an MPEG-21 environment. To demonstrate the above, a software testbed has been implemented based on MPEG-21 ISO standard for validating UMA viewer requirements by having full control of the presentation platform. In this testbed, communication between a client and a server is initiated for exchange of media content considering various factors like terminal capabilities and network conditions. The proposed streaming scheme shows the efficient exchange of MPEG-4 media, which provides a broad platform for presentation of media content over a variety of devices, in a UMA framework.

1. INTRODUCTION

The rapidly growing multimedia content on one side and the application devices on the other hand, calls for a concept which can enable the users to experience any media content on any device under different environments. The above issue is addressed in the concept of Universal Multimedia Access (UMA) [1] which deals with delivery of images, video, audio and multimedia content under various network conditions, user preferences and capabilities of terminal devices. The objective of UMA enabled system is to provide the different presentation of the same information from a single source base using appropriately chosen media conversions.

The emerging MPEG-21 standard [2, 3] enables a multimedia framework facilitating the UMA concept for transparent and augmented use of multimedia content across a wide range of networks and devices . The details of the media for exchange is encapsulated in a Digital Item (DI). This DI represents and identifies the information exchanged in a structural manner. In the MPEG-21 framework the 'User' is any entity interacting in the MPEG-21 environment. The concept of Digital Item Adaption (DIA) [4], defined in MPEG-21:part-7 provides the support for the concept of UMA. DIA specifies the tool for adaptation of the media content for various network conditions, usage environments and user preferences.

MPEG-4 [5] is the only open standard which provides easy deployment of multimedia content for any user of any device. Moreover MPEG-4 advances audio and video compression, enabling the distribution of contents and services from low bandwidths to high-definition quality across broadcast, broadband, wireless and packaged media. MPEG-4 provides a standardized framework for many other forms of media including text, pictures, animation, 2D and 3D objects which can be presented in interactive and personalized media experiences.

In this paper we present a design and streaming architecture model of the UMA viewer in MPEG-21 framework. DIA is used for supporting the UMA concept. To demonstrate the above, we have presented a client-server based software architecture model built based on the new MPEG-21 ISO standard. Here, the MPEG-4 content is used as the streaming media.

The paper is organized as follows: Section 2 brings out the UMA viewer requirements. Section 3 describes the DIA process, Section 4 explains the various parts of the proposed architecture. The demonstration setup is described in section 5 and the implementation details are presented in Section 6. Section 7 concludes the paper.

2. REQUIREMENTS FOR THE UMA VIEWER

The UMA concept will place certain requirements on the multimedia viewers, such as:

- Being able to buffer a given amount of data to prevent frame delays during small network traffic variations when the channel characteristic changes dynamically. Commercial viewers use this today; the problem is how to take control of the buffer based on dynamic channel and network feedback.
- Being able to use a media description annotation to automatically extract media conversions from an original sequence. This should be done instantaneously and continuously, or alternatively the media descriptor can be as simple as a pointer to the correct conversion of the content on the server. Whether the UMA viewer should be capable of doing media conversion or it has to be done at the server or network itself is a research problem and depends upon various factors. This is not further discussed in this paper.
- Being able to provide intelligent Quality of Service control over the streaming of the content, including fast response to changes in channel bandwidth, automated presentation of changing frame rates.
- Being able to provide adequate support for negotiation procedures during time variations in channel conditions and access schemes (also for initial setup).

Further the viewer should be able to understand the content being streamed so that it can decode the incoming bitstream. Nowadays there exists a lot of proprietary media

[1] This work was supported by the Centre for Quantifiable Quality of Service in Communication Systems, Centre of Excellence" appointed by The Research Council of Norway. http://www.ntnu.no/Q2S/

[2] The term viewer is used in a broader sense to reflect the Quality of Service driven viewing of multimedia presentations.

content which can be encoded/decoded only by the corresponding proprietary software/hardware. In such a situation the UMA viewer has no control over those specific media files. This poses problems for providing Quality of Service (QoS). To some extent this problem can be overcome if the media descriptor describes the nature of the content, which can be used to take appropriate actions for viewing the content. For example the media description can be used to invoke the correct decoder. If the media content belongs to some open standards such as ISO/IEC standards, the viewer can take control over the media content for adapting it for the present network condition, channel variations and user limitations. Further it can provide intelligent QoS control over the media.

Nowadays Internet Protocol (IP) is becoming the common denominator for multimedia services and wireless access which grows very rapidly. Multimedia over IP and wireless networks face many challenges due to the intrinsic natures of the IP and wireless networks such as unknown and dynamic bandwidth, delay jitter and packet-loss. This imposes some necessary trade-off between QoS guarantee and resources utilization efficiency. These problems should be tackled intelligently for efficient delivery of multimedia content for various Users.

At present there is no viewer which can provide all the above mentioned basic functionalities. So it becomes necessary to design a viewer incorporating all these functionalities. As a first attempt we consider the afore-mentioned requirements and propose an architecture as shown in Fig. 1. This architecture is meant for laboratory evaluation of an UMA viewer. Here the ‘reference’ is the original streamed media and the ‘enhanced’ is the stream adapted to suit the present network conditions, channel variations and user capabilities. The ‘control’ is the feed back signal to the streaming server, which requests for conversion of the media.

The ‘metrics’ window provides the information regarding the quality of the adapted content with respect to the reference content. Nowadays the concept of Universal Multimedia Experience (UME) is getting more attention, which aims at the best possible presentation of multimedia content to the end Users. Hence, in the context of streaming audio-visual presentation one would require a metric that could evaluate the perceptual quality of the content with either limited or no access to reference content [6]. Such metrics are called reduced-reference (RR) or no-reference (NR) metrics. These metrics are used to monitor and improve the presentation quality at the end User terminal. Finally the ‘metadata’ window shows the description of the reference and enhanced (adapted) media content.

3. DIGITAL ITEM ADAPTION (DIA)

The goal of MPEG-21 (ISO/IEC 21000) is to achieve interoperable transparent access to multimedia content by shielding Users from the network and terminal installation, implementation and management issues. To achieve this adaptation of digital items is required. The concept of DIA is illustrated in Fig. 2. The DIA provides the users with media content tailored to their usage environments. As shown in Fig. 2 a DI is subject to a resource adaptation engine and a descriptor adaptation engine which together produce the adapted DI. The output of the DIA engine is a modified DI that is suitable for the User.

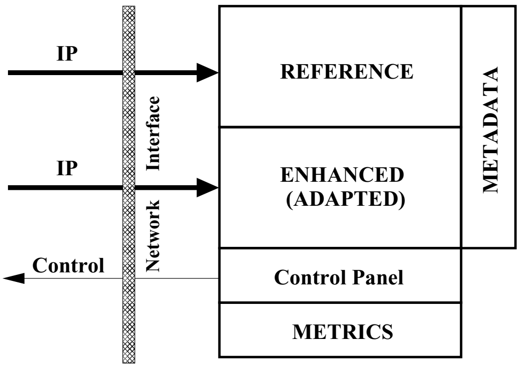

Figure 1: The UMA Viewer.

The scope of standardization lies in the DIA Tools for resource and description adaptation, and quality of service management. These DIA Tools can be clustered in to three major categories based on its functionalities.

- *Usage Environment Description Tools:* This provides information about the various dimensions of the usage environment. This includes User characteristics, terminal capabilities, network characteristics and natural environment characteristics.
- *Digital Item Resource Adaptation Tools :* These tools target the adaptation of content in a DI. The *bitstream syntax description* tools are used to facilitate the scaling of bitstreams in a format independent fashion. The *terminal and network QoS* tool describes the relationship between constraints, feasible adaptation operations satisfying these constraints and associated qualities. The *bitstream syntax description link* tool provides the facilities to create a rich variety of adaptation architectures based on tools specified within ISO/IEC standards. The fourth tool *metadata adaptability* enables the filtering and scaling of XML instances with reduced complexity as well as integration of several XML instances.
- *Digital Item Declaration Adaptation Tools :* These tools aim to adapt Digital Item Declaration as a whole. The *session mobility* enables the DI to be consumed on the second device in an adapted way. *DIA configuration* tools provides information required for the configuration of an DIA Engine.

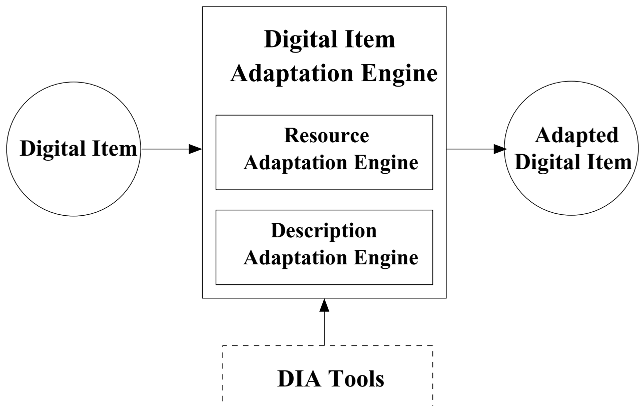

Figure 2: The Digital Item Adaptation Engine.

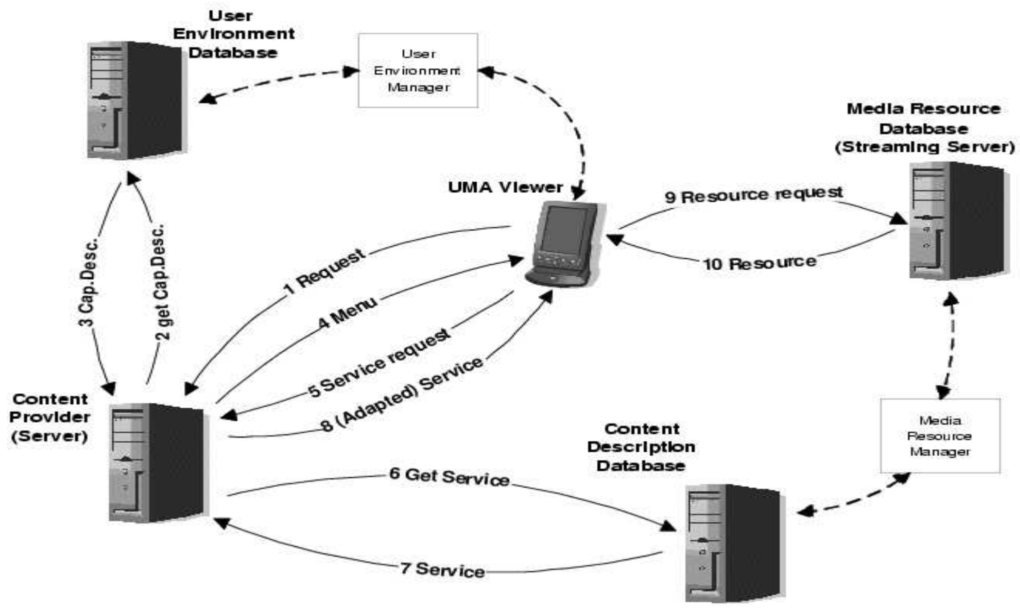

Figure 3: The proposed streaming architecture model.

4. PARTS OF THE PROPOSED ARCHITECTURE

The proposed streaming architecture is created to provide an MPEG-21 environment for streaming MPEG-4 content facilitating UMA concept. The proposed architecture contains the following five main parts:

- *User Environment Database* is responsible for maintaining a description of all the Users in the network. The description is done with MPEG-21 DIA Usage Environment Description Tools.
- *Content Provider* is responsible for gathering information from both User Environment Database and Content Description Database in order to serve the requests from the UMA Viewer in the best possible way.
- *UMA Viewer* represents the end user and is responsible for presenting the multimedia content from the content provider. It is also responsible for updating the User Environment Data-base when necessary.
- *Content Description Database* is responsible for storing and maintaining the metadata regarding the media resources that are available at the Media Resource Database. This database uses MPEG-21 DIDL (Digital Item Description Language) and DIA tools for holding the metadata.
- *Media Resource Database* is responsible for storing and delivering the actual media content. This can typically be a streaming database with RTSP-RTP/UDP support.

The User Environment Manager and Media Resources Manager are intended to be used for maintaining and updating the Databases and are to be implemented in the future work. The architecture model is shown in Fig. 3. The communication between the Server-UMA Viewer and between the Servers is done through exchange of Digital Items. For better understanding the main transactions 1,4,5 and 8 are explained below:

- *Transaction 1* The client requests for service to the Content Provider describing his Viewer limitations.
- *Transaction 4* The available services that satisfy the capability of the user are presented through a menu,
- *Transaction 5* The User makes the choice of service and requests for the service.
- *Transaction 8* The service provider adapts the content for the User and sends it to the client for viewing.

5. DEMONSTRATION SETUP

To demonstrate the proposed UMA viewer architecture, a software test-bed has been implemented. In this testbed the communication is setup between the client and the server through the respective port numbers and IP addresses. The simplified software architecture of the proposed UMA Viewer is illustrated in Fig 4 and the functionalities of some of the main classes are briefly explained below.

5.1 Console

This class is the center of the software architecture. All communication between the classes will go through this class with a few exceptions. A major issue in the design of this testbed is to have an information- flow that is easy to follow. Therefore the Console- class was selected to be the center of the testbed. The main method is also located here, so the entire program starts in this class.

Depending on the choices made by the client, different actions will take place inside this class. The methods inside this class are mainly for communication between the classes, where the 'Console' class sends the information further to the destined class.

When an element is received in the architecture, it does not know what type of element this is. Inside the 'Console' class, a method will determine the type of element received and send this element to the right parser.

When the user has set his usage environment, this is retrieved in the parsers and sent to the 'Console' class. The 'Console' then sets the content of the usage environment and

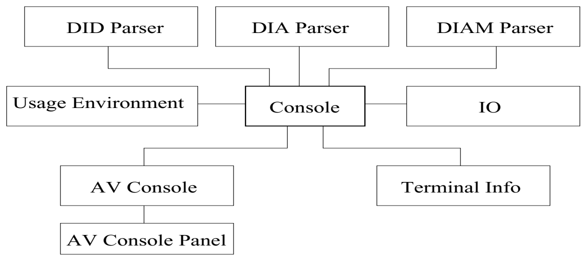

Figure 4: Simplified software architecture.

the boolean state to 'true' in the 'Usage Environment' class. If the server requires to read the information stored there, it can be reached through methods communicating with the 'Usage Environment' class. Saving elements and loading elements is done with methods in the 'Console' class. With these methods, the User can control whether the elements sent between the client and the server are correct.

When the parsers receive different elements, they will need to call methods in the other classes in the testbed. Various methods in this 'Console' class ensures that the necessary actions are taken. An example be: One of the parsers register that the client has selected "AV Presentation". The response to this choice will be that the 'AV Console' class must create an AVCDI (Audio Video Content Digital Item).

5.2 IO

IO is the interface to the outside world. This class sets up the socket that the two machines using this architecture are communicating through. Given the IP- address to the remote machine and the port number open for communication on the other machine, the socket enables the Users to send elements in between. All information sent between the Users are elements wrapped as DIDL- elements.

5.3 Terminal Info

The parameters describing the User must be known inside the architecture. Therefore, the User must set information about his terminal to run an audiovisual presentation. If a user has a very small screen, he cannot display a big image. Vice versa will it be very poor use of a terminal to play video that is small on a terminal with a big screen. Describing the terminal properly will enable matching of media content available with the description of the terminal.

The frame displayed on the screen lets the User give accurate information about the terminal. When a User has made his selections on the terminal side, these will be included in a vector containing the results. This vector will be sent to the 'AV Console' class where the information is included in the usage environment that is sent to the server.

5.4 Usage Environment

All information received on the server side about the User must be stored for further use as an element. The element will be parsed in 'AV Console' class and sent to this class through the 'set' method called from 'Console'. When the element is located here, the server can get it by calling the 'get' method from the 'Console' class. A boolean variable set to 'true' or 'false' in this class will ensure that the server knows if this element is set or not. When the server receives the response, telling that the client has selected to view a an 'AV-Presentation', the 'create AVCDI'- method in 'AV Console' will be called. This class listens to the state of the usage environment which is false initially. As the state is 'false', the method will wait for the state to be set to 'true'. When the usage environment is set, and the boolean variable is set to 'true', the 'AV Console' class will read the information stored in the element.

6. IMPLEMENTATION DETAILS

The proposed architecture uses the descriptors available in MPEG-7 [7] and MPEG-21 for streaming MPEG-4 content in such a way that Users with different capabilities can experience the content. The UMA concept has been used in the test bed through the DIA concept of the MPEG-21 standard. Implementing this testbed was done with the use of Java2 SDK. Together with the KXML parser, this SDK contains all the necessary components for an implementation. It was natural to implement this testbed with Java since it has support for so many of the features used here. Playing MPEG-4 content is done with the Quicktime- player. Setting up the player in a Java environment was done with the "QuickTime for Java" interface made by Apple. This interface lets the programmer use standard Java components and combine these with the components from Java.

7. CONCLUSION

In this paper we have addressed various issues regarding the development of an UMA viewer and proposed an architecture for UMA viewer considering all the requirements. To demonstrate this UMA capable viewer, we have designed a testbed for streaming interactive MPEG-4 media adapting to the users having various terminal capabilities and preferences given by MPEG-21 DIA descriptors. This work stands as a proof how the UMA concept can be implemented for efficient delivery of media with MPEG-21 framework.

REFERENCES

[1] A. Vetro, C. Christopoulos, and T. Ebrahimi (Editors), "Universal Multimedia Access - Special Issue," *IEEE Signal Processing Magazine*, vol. 20, no. 2, Mar. 2003.

[2] Joint Technical Committee ISO/IEC/JTC1/SC29/WG11, "Information technology multimedia framework (MPEG-21) -part 1: Vision, technologies and strategy," 2003.

[3] J. Bormans, J. Gelissen, and A. Perkis, "MPEG-21: The 21st century multimedia framework," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 53–62, Mar. 2003.

[4] IS 21000 - 7, FCD, "MPEG-21 Part-7: Digital Item Adaptation," July 2003.

[5] F. Pereira and T. Ebramhimi (editors), *The MPEG-4 Book*, IMSC Press, 2002.

[6] Odd Inge Hillestad, R. Venkatesh Babu, Ajit S. Bopardikar, and Andrew Perkis, "Video quality evaluation for UMA," in *International Workshop on Image Analysis for Multimedia Interactive Services*, Apr. 2004.

[7] B. S. Manjunath, Philippe Salembier, and Thomas Sikora (editors), *Introduction to MPEG-7 : Multimedia Content Description Language*, John Wiley & Sons, 2002.