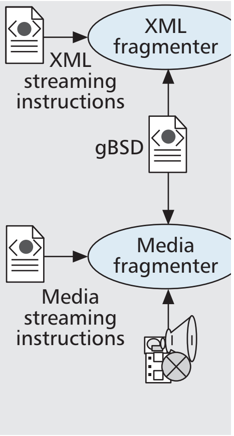
AN INTEROPERABLE DELIVERY FRAMEWORK FOR SCALABLE MEDIA RESOURCES

MICHAEL EBERHARD, CHRISTIAN TIMMERER, AND HERMANN HELLWAGNER, KLAGENFURT UNIVERSITY
EMANUELE QUACCHIO, ST MICROELECTRONICS



The authors present an interoperable framework for the delivery of scalable media resources. The framework provides support for Video on Demand as well as multicast streaming.

ABSTRACT

In this article an interoperable framework for the delivery of scalable media resources (e.g., in the standardized scalable video coding format) is presented. The framework provides support for video on demand as well as multicast streaming, and performs efficient, generic, and interoperable adaptation of streamed content based on MPEG-21 Digital Item Adaptation. The server as well as the clients of the streaming framework implement the MPEG Extensible Middleware and utilize the MPEG Query Format for querying the available media resources. The framework has been fully integrated into the VLC media player. The architecture for both VoD and multicast is presented in detail. Finally, a comparison in terms of performance of the generic MPEG-21 metadata-based adaptation approach to an SVC-specific adaptation approach is provided.

INTRODUCTION

Today the consumption of multimedia content over the Internet or other heterogeneous networks is becoming more and more important. Due to the heterogeneity of the end users' devices as well as the networks, support for *universal multimedia access* (UMA) [1] (i.e., the ability to provide multimedia content at any time on any device with suitable quality) is a very desirable goal for multimedia delivery systems. There has been a lot of effort in the multimedia research community in the last few years to ensure that multimedia content can feasibly be adapted to suit these heterogeneous environments. In video coding, the most promising standard is *Scalable Video Coding* (SVC) [2], an extension to the *Advanced Video Coding* (AVC) standard. SVC provides scalability in three dimensions, temporal, spatial, and signal-to-noise ratio (SNR) scalability, and allows a fully scalable bitstream to be created with no more than 10 percent of overhead in terms of bit rate compared to a non-scalable AVC-encoded bitstream [2].

Although a scalable bitstream allows the layers to be extracted in a convenient way, the bitstream still needs to be adapted to suit consumers' usage environments. A generic and efficient way to perform such adaptations based on XML metadata is provided by *MPEG-21 Digital Item Adaptation* (DIA) [3].

In this article a framework for the interoperable streaming of scalable multimedia content is presented. The framework not only supports the SVC extensions of AVC, but utilizes MPEG-21 DIA metadata for codec- and device-independent adaptation of scalable content according to usage environment conditions (e.g., terminal/network capabilities) signaled by the receiving terminal. These usage environment conditions — conforming to MPEG-21 DIA — are encapsulated within the *Request Content Protocol* (RCP) defined in *MPEG Extensible Middleware* (MXM) [4]. MXM provides the means for an application programming interface (API) for interoperable access to various MPEG technologies (e.g., video/audio coding formats, file and transport formats, and various description formats), which shall enable accelerated development and deployment of new digital media products, applications, and services. The MXM API and a reference implementation are publicly available; for further information the interested reader is referred to [4]. For the initial request (i.e., to query which media resources are available), we adopted the *MPEG Query Format* (MPQF) [5]. Hence, with the usage of the MPQF, MXM, MPEG-21 DIA, and SVC we present a fully interoperable solution to the UMA problem while preserving efficiency and considering performance.

The remainder of this article is organized as follows. In the next section an introduction to SVC is given. In the following section the adaptation of scalable content utilizing MPEG-21 DIA metadata is presented in detail. We then describe the video on demand (VoD) architecture, which allows each client to receive a separate bitstream at suitable quality. The corresponding multicast architecture, which provides one or more layered bitstreams to all of the clients, is then presented. We then provide a comparison of the MPEG-21

 1536-1284/09/$25.00 © 2009 IEEE

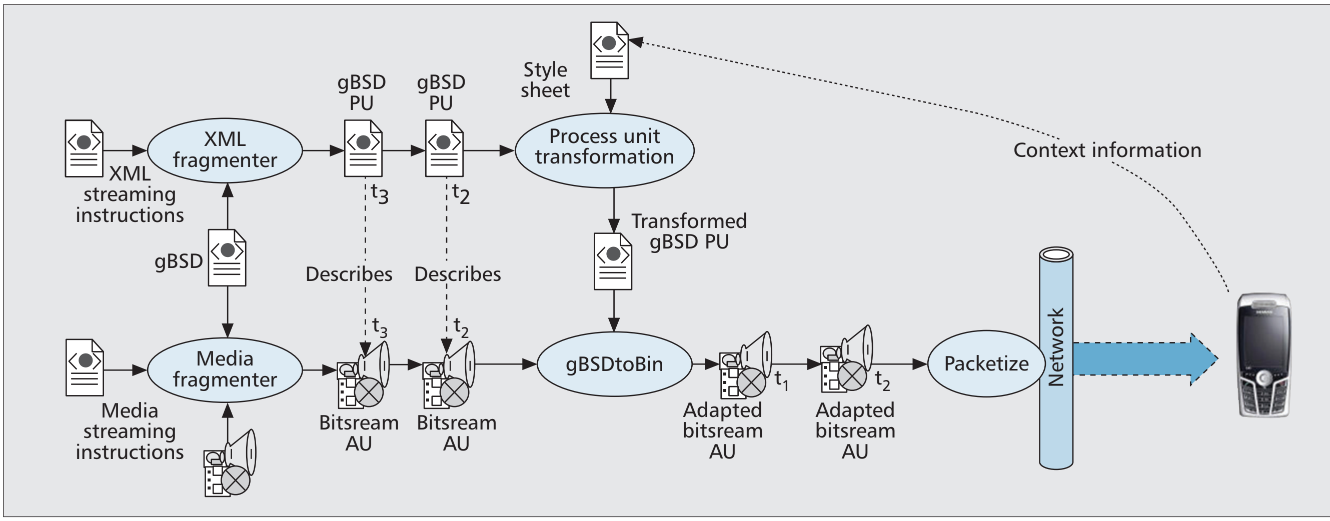


Figure 1. *MPEG-21 DIA metadata-based adaptation of scalable content in a streaming scenario.*

DIA metadata-based adaptation approach vs. an SVC codec-specific adaptation approach. The final section concludes the article.

Scalable Video Coding

The SVC extensions of the AVC standard provide different layers within one bitstream. The standard defines three different types of scalability: temporal scalability, which allows the selection of different frame rates; spatial scalability, which allows different resolutions to be provided; and SNR or quality scalability, which allows the visual quality to be influenced. One main advantage of the SVC standard compared to other standards is that the base layer, which represents the minimum quality of the bitstream, is compatible with AVC and can be processed by any AVC-conforming decoder.

As specified in the AVC standard, SVC utilizes the *network abstraction layer* (NAL) and *video coding layer* (VCL) to structure the SVC content. Thus, all the data of an SVC bitstream is stored in NAL units, which can be either VCL NAL units containing data representing video samples or non-VCL NAL units that contain other data like parameters or *supplemental enhancement information* (SEI). An *access unit* (AU) contains all the VCL and non-VCL NAL units for a single frame. The most important parameters for SVC bitstreams are the *sequence parameter set* (SPS) and picture parameter set (PPS). While the SPS contains parameters that remain constant for the whole video sequence, the PPS contains parameters that might change for every picture. In SVC, every layer has corresponding SPS and PPS parameters.

If an SVC bitstream is adapted, each AU has to be processed, and those NAL units not needed for the desired quality are truncated. Additionally, some minor adjustments like updating the number of SPS or PPS parameters have to be performed. Compared to other adaptation approaches like transcoding, these truncation and minor adjustment operations can be performed in a very efficient way and implemented on most devices.

Adaptation of Scalable Content

To support the idea of UMA, a variety of scalable codecs have been developed in recent years. Although the adaptation of scalable bitstreams can be performed efficiently, the adaptation process still has to be performed by a suitable adaptation engine. However, most of today's adaptation engines for scalable codecs are codec-specific and hence require using different adaptation engines for different scalable codecs. In Part 7 of the MPEG-21 standard, DIA, a metadata-based adaptation approach, is defined that allows adapting all scalable content utilizing the same adaptation engine (i.e., codec-independent adaptation). The adaptation process utilizing MPEG-21 DIA within a streaming scenario is illustrated in Fig. 1.

To prepare the scalable bitstream for adaptation, first a high-level metadata description of the syntax of the bitstream, the so-called generic *bitstream syntax description* (gBSD), has to be created. The gBSD describes the bitstream by means of *gBSDUnits*, which describe content portions that can either be kept or truncated, and *Parameters*, which describe content portions that can be changed during the adaptation process. Although the gBSD is basically sufficient to adapt the bitstream, it has to be considered that the properties of the usage environments of clients can change during the streaming process. Thus, the adaptation of the bitstream has to be performed in a timed manner, to ensure that changes of clients' usage environments are applied to the adaptation process in a timely manner. To support such timed adaptation, the so-called *streaming instructions*, which describe the timing of the bitstream's AUs as well as the parts of the gBSD, are added to the gBSD. While the *XML streaming instructions* (XMLSI) describe how the gBSD should be fragmented into the so-called gBSD *process units* (PUs) and how the transmission of the gBSD process units should be timed, the *Media Resource Streaming Instructions* (MRSI) allow the AUs to be localized within the bitstream and provide the timing information for those AUs.

The adaptation process is performed as follows. The XMLSI in the gBSD are utilized to extract the PUs from the gBSD, and the MRSI

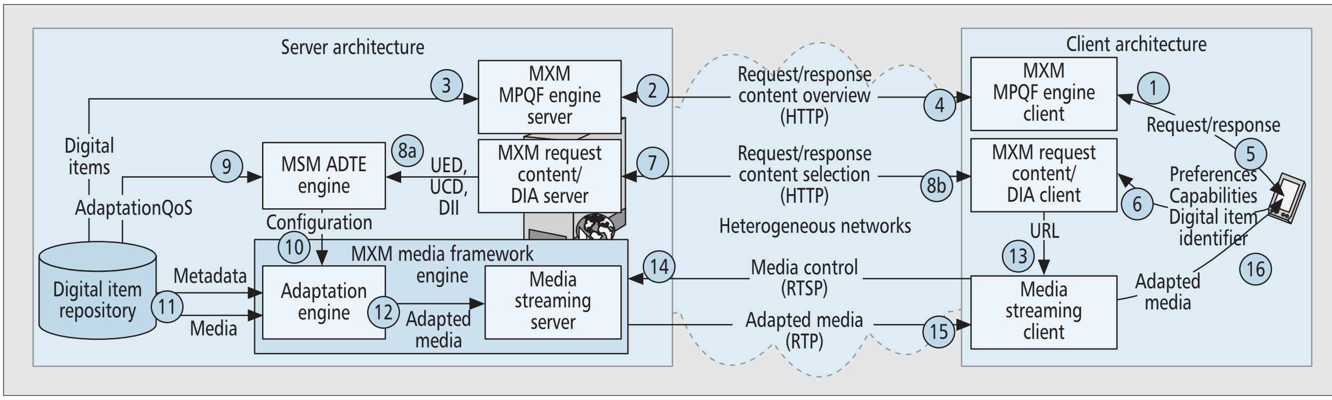


Figure 2. *Interoperable architecture for video on demand.*

are used to extract the access units from the bitstream. The PUs as well as the access units are forwarded in a timed manner to ensure that the changes are immediately visible if the adaptation parameters are changed. The actual adaptation process consists of two steps. First, the gBSD PUs are transformed utilizing adaptation parameters. During this transformation process, the gBSDUnits of the NAL units that are not needed for the adapted bitstream are truncated, while the Parameter elements that need to be changed during the adaptation process are adjusted. This transformation is based on a style sheet, which describes how the gBSD for a specific codec needs to be transformed. The commonly used way to perform the style sheet transformation is by using an *Extensible Stylesheet Language for Transformations* (XSLT) library. Second, after the transformation of the gBSD PUs, the transformed PUs are used to extract the adapted access units. This adaptation of the access units is performed by the *gBSDtoBin* process. The resulting adapted access units are finally streamed to the user. As both the XSLT transformation and the gBSDtoBin process are normatively specified in their respective standards, the actual adaptation of the scalable bitstream is performed solely based on metadata (i.e., gBSD and XSLT stylesheet) and hence is coding-format-independent.

Interoperable Architecture for VoD

As a first scenario, a VoD testbed was implemented. The testbed consists of one server and a number of possibly heterogeneous clients. The VLC media player provides the streaming capabilities for both the server and the clients, and has been extended by additional codecs to support SVC. To enable the setup of VoD sessions based on clients' usage properties, the VLC on the server and client sides has been extended with an *MXM MPQF engine* and an *MXM request content engine*. Additionally, the VLC server has been extended with an *MXM adaptation decision-taking engine* (ADTE) and an *MXM media framework engine* to support the adaptation of scalable bitstreams. The detailed procedure of how such a VoD request is handled by our customized VoD server and client is illustrated in Fig. 2 and described in the following.

To start the VoD session, the client first needs to request the available video sequences from the server by instructing the MXM MPQF engine to construct a query. The MPQF-conforming input query is transmitted to the server using HTTP and is received by the MXM MPQF engine at the server. The engine subsequently examines the *digital item repository* for available items and transmits an MPQF-conforming output query containing all available sequences to the client. After the receipt of the output query from the client, the user selects one of the sequences and sets the adaptation preferences for the VoD session. This information is forwarded to the MXM Request Content Client, and an MXM Request Content message containing the identifier of the selected sequence and the properties of the usage environment is created. The properties of the usage environment are expressed utilizing the *MPEG-21 DIA usage environment description* (UED) and *universal constraint description* (UCD) [3]. While the UED allows expressing properties such as user characteristics, terminal characteristics, network characteristics, or properties of the natural environment, the UCD allows further constraints to be imposed on the properties of the UED. The MXM Request Content message containing the UED/UCD is transmitted to the server by means of HTTP. The MXM Request Content engine at the server responds with an MXM Request Content Response message that contains a personalized *Real Time Streaming Protocol* (RTSP) URL that can subsequently be used by the client to start the streaming session tailored to its capabilities.

To support these functionalities on the client side, an MXM user interface including the MXM MPQF engine, the MXM request content engine, and an optimized SVC decoder have been integrated into the VLC media player. The MXM user interface requests the available sequences from the server and presents them to the user. The user can select the desired sequence and set the adaptation parameters (i.e., width/height, frame rate, bit rate) for the VoD session. In real-life applications, the adaptation parameters might be set automatically according to the capabilities of the user's terminal or the bandwidth of the connected network.

At the server, the MXM interface, including the MXM MPQF engine, the MXM request content engine, and the MXM ADTE, handles requests from clients. After receipt of the MXM Request Content message, the decision of how to adapt the SVC bitstream needs to be made. Thus, the UED/UCD are extracted from the request

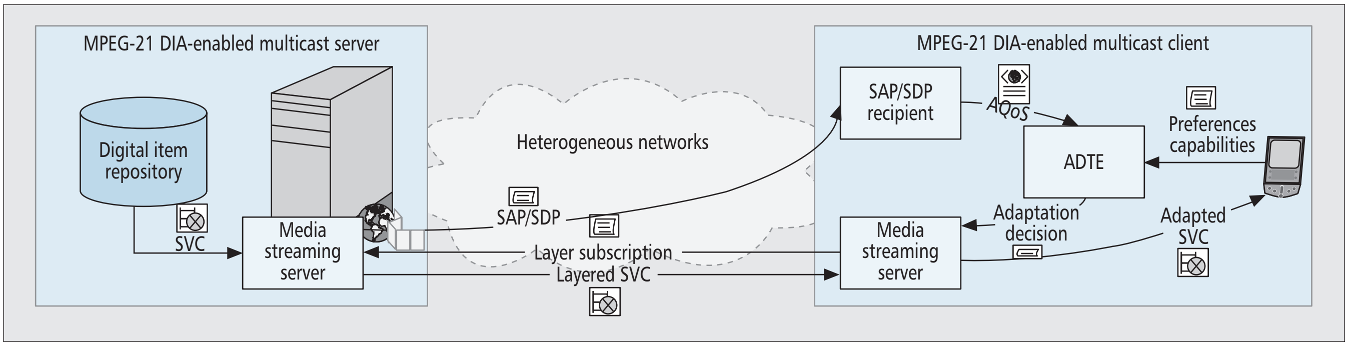


Figure 3. *Interoperable architecture for multicast streaming [7].*

message and forwarded to the MXM ADTE. To make the adaptation decision, the ADTE also accesses the *adaptation QoS* (AQoS) [3] description of the desired bitstream from the digital item repository. The AQoS describes the layers of the SVC bitstream and allows the properties of the usage environment to be matched with the available adaptation capabilities of the bitstream. The resulting adaptation parameters are stored as global VLC parameters to make them easily accessible during the streaming process.

The actual adaptation is performed during the streaming process. Every access unit is forwarded to the adaptation engine, which has been integrated into the VLC as a packetizer module. The choice to implement the adaptation engine as a packetizer was made because of its position during the processing of multimedia content. The packetizer receives the multimedia data after demuxing and before the data is encapsulated into RTP packets. Thus, the adaptation engine does not have to consider the file format or try to keep the RTP packet structure; it only has to focus on the adaptation of the scalable content. The adaptation engine performs adaptation based on the gBSD PUs as described in the previous section, and the globally stored adaptation parameters are used as parameters for style sheet transformation. As each AU is adapted independently, and the AUs are provided to the packetizer in a timed manner, the adaptation parameters can be changed dynamically. Thus, when the client wants to change the adaptation parameters, the request content message is sent again to the MXM interface utilizing the session identifier, the adaptation decision is made again, and the globally stored adaptation parameters are updated. On receipt of the next access unit, the update of the parameters is visible at the client.

To demonstrate that the interoperable delivery framework is also well suited for accessing content on mobile devices, the client has been ported to ST-Ericsson's STn8810-Nomadik platform [6]. For the porting, the VLC has been cross-compiled for ARM processors, and an ARM-optimized AVC decoder plug-in has been integrated into the VLC instead of the optimized SVC codecs. As the SVC base layer is fully compatible with AVC, the client on the mobile platform could successfully receive and decode the bitstreams at base layer quality. Finally, please note that the mobile platform used for the porting of the framework (i.e., STn8810) has been replaced by newer models (e.g., STn8815, STn8820) in the meantime.[1]

Interoperable Architecture for Multicast

While the VoD scenario aims to provide each client with an individual bitstream tailored to the client's usage environment, the multicast scenario aims to save bandwidth by providing the same layered bitstream to all clients. The testbed of the multicast scenario consists again of one multicast server and a number of multicast clients. For each SVC layer, a separate RTP multicast session is started, and each of the layers can be subscribed to by clients depending on their usage environment properties.

The information about the properties of the SVC layers is provided to all the clients in the format of the *Session Description Protocol* (SDP) and is distributed utilizing the *Session Announcement Protocol* (SAP). In addition to the information about the streaming sessions, the SDP description contains the SPS and PPS parameters, which are encoded in base64. Furthermore, the scalability parameters (i.e., the resolution, frame rate, and bit rate) are provided within the SDP description. These parameters are subsequently used by the clients to decide to which of the layers to subscribe. The architecture of the multicast scenario is illustrated in Fig. 3.

The *multicast server* continuously broadcasts the SDP description utilizing SAP to all of the multicast clients. The SDP description contains all the information needed to start the streaming process with the desired layers. Furthermore, the server performs a multicast of all the scalable layers in separate RTP sessions.

The *multicast client* stores the information of all the received SDP descriptions to create a playlist and allow the user to choose the preferred stream. After the selection of the bitstream, the information about the layers of the bitstream is extracted from the SDP description. The information about the layers is utilized to create the AQoS description. The preferences and usage environment properties that are locally available are formatted as UED/UCD and are utilized together with the AQoS description by the ADTE to make the adaptation decision in exactly the same way as is done on the server in the VoD scenario. Based on the resulting adaptation parameters, the client subscribes to the appropriate layers of the SVC bitstream, and the streaming of the multimedia data is started.

[1] *http://www.stericsson.com/platforms/nomadik.jsp*

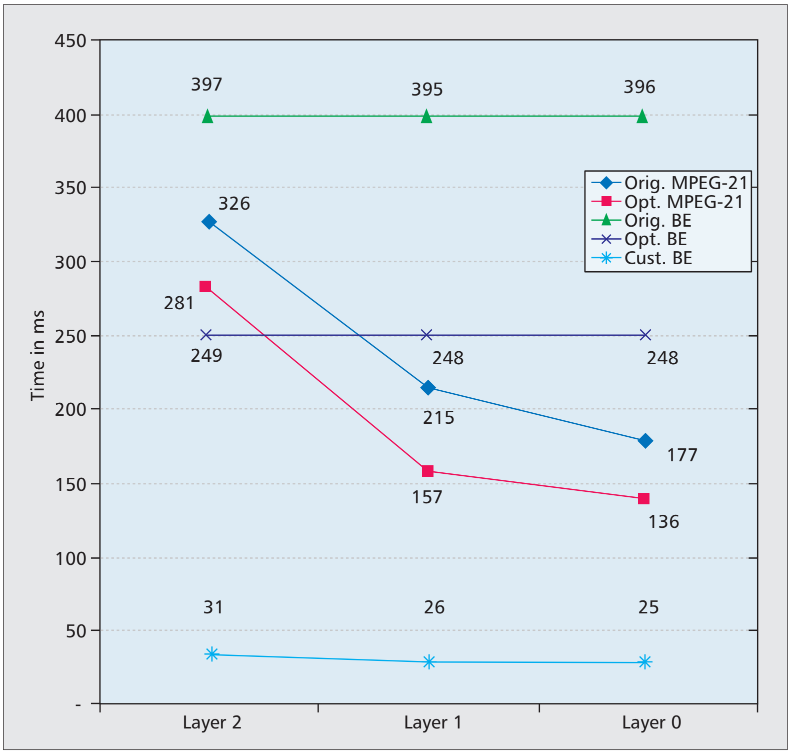


Figure 4. *Performance comparison for sequence ice.*

Comparison to Codec-Specific Adaptation

To evaluate the performance of our codec-independent MPEG-21 DIA metadata-based adaptation approach in comparison to an SVC-specific adaptation approach, the metadata-based approach has been compared to the *BitstreamExtractor* of the JSVM reference software 9.1 [8]. To perform a feasible comparison of the two adaptation approaches, the offline adaptation of a number of sequences has been measured. First measurements have shown that the metadata-based approach clearly outperforms the reference software [9]. The main reason for the performance gain of the metadata-based approach is that the gBSD clearly indicates which parts of the bitstream need to be copied. Thus, only those parts of the bitstream that are actually needed for the adapted bitstream are processed. On the other hand, the JSVM reference software always needs to parse the complete bitstream to locate the NAL units needed for the adapted bitstream. This behavior results in poor performance compared to the metadata-based approach if adaptation to a lower quality variant is performed.

Additionally, the performance measurements have been utilized to identify the bottlenecks for both approaches and to optimize them [7]. For the metadata-based adaptation approach, the style sheet transformation was identified as the main bottleneck. Thus, alternatives to the traditionally used XSLT libraries have been investigated. After some evaluations, the usage of a generic *libxml*-based[2] adaptation interface has been found to be the fastest way to transform the XML metadata documents while still keeping the transformation process codec-independent. For the BitstreamExtractor application, the processing of the complete bitstream, no matter which parts of the bitstream were required, has been identified as the main reason the BitstreamExtractor showed poorer performance than the metadata-based adaptation approach. As the NAL unit headers provide only the scalability parameters but no length information, the BitstreamExtractor has to parse even NAL units that are not needed for the adapted bitstream, because it needs to find the start code for the next NAL unit. To avoid this overhead, two improvements were implemented. First, length information has been added to the NAL unit header, which makes sure that the BitstreamExtractor only has to parse the headers and needs to process only those parts of the bitstream which are really required for the adapted bitstream. Second, a customized and lightweight BitstreamExtractor application has been implemented, which does not aim to provide the full functionality of the JSVM reference software's BitstreamExtractor but just tries to adapt the bitstream as efficiently as possible. The results of the initial as well as optimized implementations are depicted in Fig. 4 for the sequence *Ice*, which contains a base layer and three enhancement layers. The results are shown for one sequence only, but the measurements have been performed for a number of sequences and all results have been very similar. In the figure, *Layer 2* indicates the time it takes to generate an adapted bitstream including all layers up to (and including) layer 2, while *Layer 0* indicates the time it takes to generate an adapted bitstream with the lowest quality (i.e., containing only the base layer). *Layer 3* is not shown as it represents the highest quality and is available without applying any adaptation.

The results for the original implementations of the JSVM reference software and the metadata-based approach confirm what has been mentioned before: the metadata-based approach shows clearly better performance when an adaptation to a lower quality level is performed. This is mainly due to the behavior of the gBSDtoBin process, which only needs to process those data needed for the adapted bitstream, while the BitstreamExtractor always has to parse the entire bitstream. Compared to the original implementations, the performance for both optimized implementations has been significantly improved. For the metadata-based adaptation approach, this improvement is due to the usage of the generic libxml-based adaptation interface instead of an XSLT library. For the BitstreamExtractor, the improvement is due to the usage of length information in the NAL unit headers. As the BitstreamExtractor can utilize the length information to directly access the NAL unit headers instead of parsing the bitstream until the NAL unit start code is found, the performance improves. However, the JSVM reference software's implementation of the BitstreamExtractor is not intended to utilize length information, which leads to constant performance of the adaptation process even if an adaptation to lower quality levels is performed. The full benefit of the usage of length information in the NAL unit headers is exploited by the customized BitstreamExtractor, which provides an optimized and lightweight implementation. The customized BitstreamExtractor clearly outperforms all other

[2] *http://xmlsoft.org/*

implementations, and its performance improves if adaptation to a lower quality variant is performed. However, the customized BitstreamExtractor performs a codec-specific adaptation of SVC content and does not provide the full functionality of the JSVM reference software's BitstreamExtractor.

CONCLUSION

In this article an interoperable framework for the streaming of scalable multimedia content has been presented. The framework allows the delivery of scalable content to users tailored to their preferences and the capabilities of their terminals. The adaptation of the content is performed in an efficient and codec-independent way by utilizing MPEG-21 DIA metadata, implementing the MXM, and adopting the MPQF. The framework provides support for both VoD and multicast streaming. On one hand, the VoD implementation allows each client to consume a separate bitstream at the desired quality. The desired quality is communicated to the server by utilizing the MPEG-21 DIA UED and UCD. On the other hand, the multicast implementation provides a layered stream arrangement to all clients. For this scenario the client achieves the desired quality by subscribing to the suitable layers, which are described by the SAP/SDP announcement. The decision of to which layers to subscribe is again made based on MPEG-21 DIA metadata. Furthermore, the porting of the framework's client to ST-Ericsson's Nomadik platform has been described to demonstrate that the framework works well with mobile devices.

The results of the performance evaluations show that the codec-independent metadata-based implementation performs significantly better than the codec-specific implementation of the JSVM reference software if an adaptation to lower quality levels is desired. Furthermore, bottlenecks for both adaptation approaches have been identified and optimizations of both approaches have been proposed. While the generic libxml-based transformation interface significantly improves the performance of the metadata-based adaptation compared to the usage of an XSLT library, the usage of length information in the NAL unit headers improves the performance of the BitstreamExtractor as it does not have to process NAL units that are not needed for the adapted bitstream. However, the implementation of the optimized and lightweight BitstreamExtractor shows that there is still a lot of potential for further improvements in both approaches, although the optimized BitstreamExtractor does not provide the codec independence of the metadata-based adaptation or the full functionality of the JSVM reference software.

ACKNOWLEDGMENT

This work is supported in part by the European Commission in the context of the P2P-Next project (FP7-ICT-216217). Further information is available at http://www.p2p-next.org/

REFERENCES

[1] R. Mohan, J. R. Smith, and C.-S. Li, "Adapting Multimedia Internet Content for Universal Access," *IEEE Trans. Multimedia*, vol. 1, no. 1, Jan.–Mar. 1999.

[2] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Trans. CSVT*, vol. 17, no. 9, Sept. 2007, pp. 1103–20.

[3] A. Vetro, "MPEG-21 Digital Item Adaptation: Enabling Universal Multimedia Access," *IEEE Multimedia*, vol. 11, no. 1, Jan.–Mar. 2004, pp. 84–87.

[4] MPEG Extensible Middleware; http://mxm.wg11.sc29.org

[5] M. Döller *et.al*., "The MPEG Query Format: Unifying Access to Multimedia Retrieval Systems," IEEE *Multimedia*, vol. 15, no. 4, Oct.–Dec. 2008, pp. 82–95.

[6] STn8810 Data Brief; http://us.st.com/stonline/products/literature/bd/11196/stn8810.htm

[7] M. Eberhard *et al*., "An Interoperable Streaming Framework for Scalable Video Coding Based on MPEG-21," *Proc. VIE '08*, Xi'an, China, July 2008.

[8] Joint Scalable Video Model (JSVM) 9, "Joint Video Team (JVT) of ISO/IEC and MPEG & ITU-T VCEG, N8751," Marrakech, Morocco, Jan. 2007.

[9] M. Eberhard *et al*., "Performance Analysis of Scalable Video Adaptation: Generic versus Specific Approach," *Proc. WIAMIS '08*, Klagenfurt, Austria, May 2008.

The adaptation of the content is performed in an efficient and codec-independent way by utilizing MPEG-21 DIA metadata, implementing the MXM, and adopting the MPQF.

BIOGRAPHIES

MICHAEL EBERHARD (michael.eberhard@itec.uni-klu.ac.at) received his M.Sc. (Dipl.-Ing.) from the University of Klagenfurt in March 2008. In his Master's thesis, he evaluated approaches to perform adaptation of scalable bitstreams in streaming environments utilizing the MPEG-21 multimedia framework. In January 2008 he started to work at the Institute of Information Technology (ITEC) at the University of Klagenfurt in the scope of the P2P-Next project. Since 2008 he has published several papers on multimedia adaptation in streaming environments. The current focus of his work is the integration of an adaptation mechanism for scalable bitstreams into P2P systems.

CHRISTIAN TIMMERER (christian.timmerer@itec.uni-klu.ac.at) received his M.Sc. (Dipl.-Ing.) in January 2003 and his Ph.D. (Dr.techn.) in June 2006 both from the University of Klagenfurt. He is currently an assistant professor at ITEC/Multimedia Communication. His research interests include the transport of multimedia content, multimedia adaptation in constrained and streaming environments, distributed multimedia adaptation, and quality of service/quality of experience. He has published more than 50 scientific papers in these areas, and he was the general chair of WIAMIS 2008. Additionally, he is an editorial board member of *IEEE Computer Science Computing Now*. He has actively participated in several EC-funded projects, notably DANAE, ENTHRONE, and P2P-Next. Furthermore, he participated in the work of ISO/MPEG for several years, notably in the areas of MPEG-21, MPEG-M (MXM), and MPEG-V.

EMANUELE QUACCHIO (emanuele.quacchio@st.com) received his Diploma and Master's degrees in electronic engineering from the Politecnico di Torino, Italy, in 1999 and 2003. He worked for two years as a researcher under a grant in the Department of Electronics of the same university and joined ST Microelectronics in the AST system R&D group in 2006. Since 2002 he has worked on multimedia related activities, in particular in image and video compression standards (JPEG, MPEG2/4, H.264/AVC, SVC, MVC) and multimedia streaming technologies and protocols. He has published and co-authored several papers in the principal journals of engineering and conferences, and has taken part in Italian and EU-IST research projects. He has given lectures at Politecnico di Torino on video coding and video streaming technologies.

HERMANN HELLWAGNER [M] (hermann.hellwagner@itec.uni-klu.ac.at) is a full professor of computer science at the Institute of Information Technology, University of Klagenfurt, Austria. He is head of the Multimedia Communication (MMC) research group. His current research areas are distributed multimedia systems, multimedia communications, and Internet QoS. His group has contributed actively to the MPEG-21 standardization effort, such as multimedia bitstream description techniques within MPEG-21 Digital Item Adaptation (DIA). He is the editor of several books, has published widely on parallel computer architecture and parallel programming, and now publishes regularly on multimedia communications and adaptation in refereed journals and conference proceedings. He is a member of ACM, GI, and OCG as well as deputy head of the Austrian delegation to the Moving Picture Experts Group (MPEG — ISO/IEC JTC1/SC29/WG11).