# Targeting Heterogeneous Multimedia Environments with Web Services

Stijn Decneut
IMEC
Kapeldreef 75
Leuven, Belgium
stijn.decneut@imec.be

Filip Hendrickx
IMEC
Kapeldreef 75
Leuven, Belgium
filip.hendrickx@imec.be

Steven Van Assche
VRT
Auguste Reyerslaan 52
Brussels, Belgium
steven.vanassche@vrt.be

Lode Nachtergaele
IMEC
Kapeldreef 75
Leuven, Belgium
lode.nachtergaele@imec.be

## Abstract

*The number of networked multimedia platforms that are introduced into the market has increased dramatically in recent years. Current approaches to multimedia distribution do not scale to this growing set of client configurations and heterogeneous dynamic networks. We propose a distributed architecture that offers a scalable solution to multimedia publication and distribution in such heterogeneous environments. It builds upon recent standardization efforts related to web services. This paper details the multimedia web services at the proxy server, that cooperate on a loosely coupled basis to tailor content creators' multimedia presentations to clients' environments. The experiments show that our web service-oriented architecture offers a significant added value in heterogeneous multimedia environments.*

## 1. Introduction

Large-scale production and distribution of media has traditionally been the private playground of large media corporations. The limited amount of widespread broadcast and display standards (such as PAL and NTSC) enables these companies to reach large audiences at a very low cost per consumer [16]. Every media company can safely assume that almost everyone is able to receive and consume its media content when distributed according to the standards.

However, with new multimedia consumption platforms and standards being introduced into the market at a high rate, each one with specific capabilities, massive challenges are arising [17] [37]. Due to the inherent mobile nature of these platforms, their capabilities and connecting network environment may even change dramatically at runtime [4]. Content creators are looking for ways to cost-efficiently publish their content in this heterogeneous environment, where they have to distribute content over multiple dynamic networks to various consumption platforms. The simplest way for content creators to target such an environment is to place their multimedia content on a server in a number of formats that suit popular network and platform scenarios. This approach, often referred to as 'simulcast', has been adopted widely on the Internet for audio and video presentations. It suffers from serious drawbacks, however, as it doesn't scale to large sets of scenarios, and is unable to cope with dynamically changing consumer environments.

A number of scalable multimedia formats have been introduced to alleviate these problems [3] [13] [14]. Still, despite their complexity, they cannot anticipate the heterogeneous set of environments and all the (proprietary) multimedia formats that will be used in the near future [38].

We propose to introduce a web service-oriented architecture with transmoding services in the network, at the proxy server, to offer a scalable approach to publishing multimedia to a heterogeneous environment. With transmoding, we take a broader view to multimedia adaptation than traditional transcoding. A transmoded multimedia item may have a very different appearance than its original, e.g. a textual transcript of an audio sample containing speech or a bitmap version of a vector image. Our service-oriented architecture takes the responsibility of tailoring multimedia data to a suitable format for consumer environments. It provides for continuous adaptation of multimedia presentations and items to the changing environments in which users wish to consume them. Even when the capabilities of clients change dramatically at run-time, our architecture continues to adapt multimedia to well-suited formats.

The key added value of our architecture over similar initiatives (see Section 5) is that we use recent standardization efforts like SOAP [34], WSDL [32] and DIME [18] extensively to build a loosely coupled, open, flexible and scalable architecture.

In Section 2, we describe three families of strategies for run-time adaptive multimedia publication. The section provides a background for situating our research. We describe our architecture in detail in Section 3. To illustrate the applicability of our approach, we present a case study in Section 4. Related research activities and architecture frameworks are discussed in Section 5.

## 2. Approaches to Run-Time Adaptive Multimedia Publication

There are currently three popular approaches to delivering multimedia to multiple platforms.

One way is to enforce a 'common-denominator' standard, like FM for radio and PAL or NTSC for TV images. While adhering to these standards, a platform may offer even better capabilities than the standards require (e.g. higher screen resolution). Yet, no matter what the capability of a platform is, it will only render the received multimedia according to the specified standards. For instance, a High Definition Television set may be capable of displaying images at a much higher resolution and frame rate than those defined by the standard, nevertheless it will only display images according to the standard's specifications.

The differences between client platforms are often more significant in a network context than in the traditional broadcasting world. A standard that is developed specifically for one platform clearly doesn't scale to very heterogeneous consumer environments where display resolution, processing power and memory size differ tremendously and multimedia data becomes ever more complex and diverse. In such environments, multimedia needs to be made available in specific encoding and formats suitable for the target consumer environments depending on their available resources. Therefore, the Internet today follows a different approach to the delivery of multimedia. A multimedia presentation is typically placed on a server in multiple versions, each one targeting a popular network connection speed and multimedia player configuration. Some examples of very popular versions are 56kbps, 100kbps and 300kbps versions of Windows Media [2] and Real Media [3]. Naturally, web pages that are developed for specific browsers and screen resolutions (e.g. 800 x 600 pixels) can also be regarded as such versions of multimedia presentations.

Other interesting approaches are proposed by MPEG-2 and MPEG-4. They describe a layered approach to video encoding, allowing one multimedia presentation to scale to different bandwidths. On top of a base layer, which contains encoded media that every client should be able to receive and decode, reside several enhancement layers with extra information that can be consumed by clients with higher bandwidths and decoding capabilities [25]. Such an approach is very suitable for highly responsive adaptation of fairly simple multimedia (audio and/or video), within a limited range of capability changes.

The first two approaches are examples of *device-specific authoring*. Device-specific authoring doesn't scale to the very large set of client configurations and heterogeneous dynamic networks that multimedia distributors will have to address in the near future [5].

The scalable approaches can be described as *multi-device authoring*. Even though they provide excellent results when used for a limited range of consumer environments in specific situations, it is impossible to anticipate all modifications that need to be made when complex multimedia content (multiple audio, video, text and other items) needs to be published to a very heterogeneous set of consumer environments.

In those situations *automatic re-authoring* may prove to offer a more elegant solution. Automatic re-authoring is based on a software system that analyses a multimedia presentation together with the characteristics of the target environment and transforms the presentation (and the items therein) so that it can be transported efficiently and rendered appropriately on the target device. The re-authoring software system is often placed on a proxy server, as proposed by [5] [8] [10].

Automatic re-authoring is particularly interesting when consumer environments change within a session. Such run-time changes often occur in mobile environments where sudden drops in bandwidth or processing power may occur at unpredictable moments in time. Whereas device-specific authoring can not handle any run-time changes (a massive amount of versions would be required), multi-device authoring is only suited for a particular limited range of changes. Automatic re-authoring allows full tailoring of multimedia to specific client preferences, device capabilities and available network at run-time. It is often combined with various caching algorithms to limit the amount of re-authoring and transmoding work that needs to be performed [17]. Naturally, automatic re-authoring can easily be combined with the other two approaches, as our architecture illustrates.

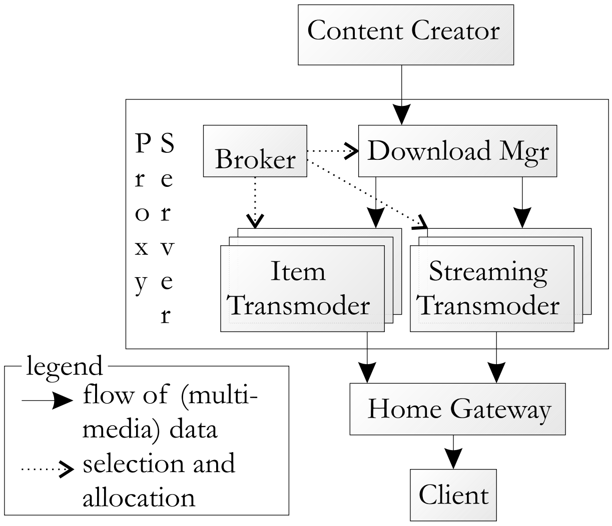


**Figure 1. The web services at the proxy server adapt the Content Creator's multimedia and place it on the consumer's Home Gateway.**

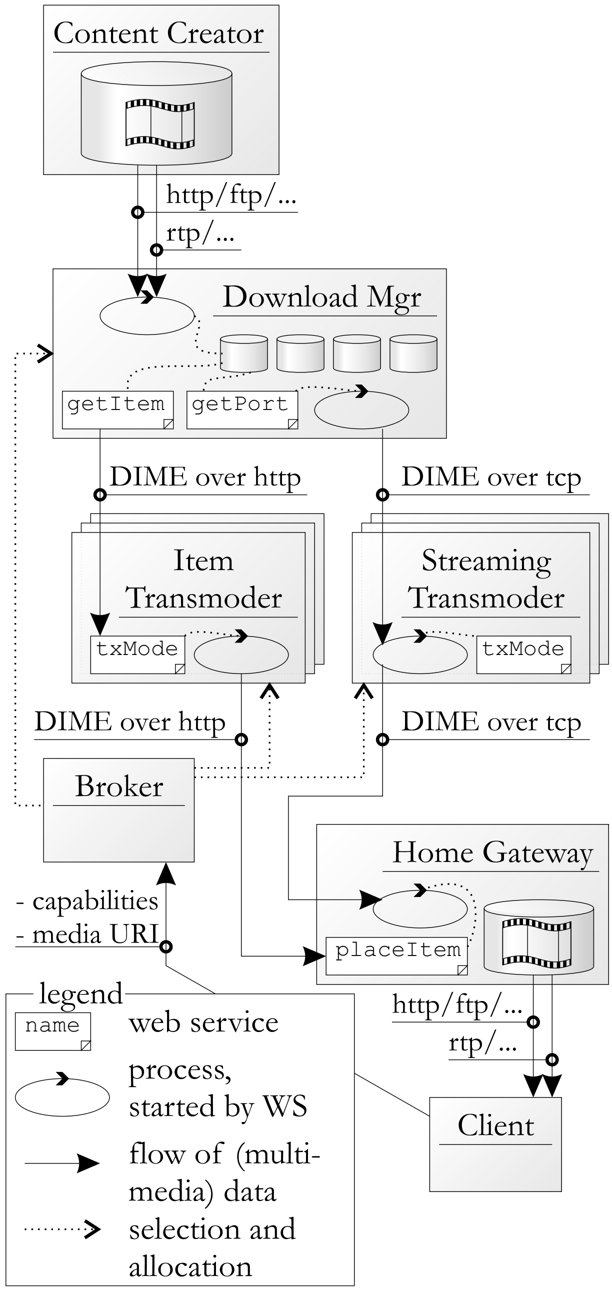


**Figure 2. A more detailed look at the services (and the communication in between) that adapt the Content Creator's multimedia to the capabilities of the Client.**

## 3. A Web Service-Oriented Architecture

Figure 1 illustrates our service-oriented architecture. We introduce several multimedia web services at the proxy server, that cooperate on a loosely coupled basis to tailor content creators' multimedia to clients' environments: the *Download Manager* (Section 3.4), *Transmoder* (Section 3.5) and *Broker* (Section 3.2) services. Another multimedia web service runs on the *Home Gateway* (Section 3.3), offering the Transmoder services easy procedures to provide their results to the client.

As stateless SOAP-based web services do not provide all required functionalities for our multimedia web services, we specify some elementary extensions in Section 3.1.

Using our service-oriented architecture, content creators may provide multimedia items and/or presentations using the formats and protocols they prefer. We assume that several formats and protocols will remain in use in the future. Even though we do not require the use thereof, we advise to use open standard formats for multimedia items, like SVG [30] for vector graphics and SMIL [31] for presentations, since they have the advantage of widespread tool-support. Furthermore, content creators are encouraged to provide their multimedia presentation in a generic and open format, like XiMPF [23] [24] or XHTML [29] and CSS [26], to enable maximum flexibility for the Transmoder services. Apart from formats and descriptions, a content creator may provide access points to its multimedia over any combination of transport protocols like HTTP and RTP [22].

A more detailed view on the architecture is depicted in Figure 2. The client indicates to a Broker service which multimedia presentation it wishes to consume and what its current capabilities are (using CC/PP). Upon this request, the Broker service contacts Download Manager services to provide the required multimedia items. The Download Manager services preferably offer these items in a popular, widespread, generic and high quality format, to facilitate the re-authoring process later on. While the Download Manager services are retrieving (from the content creator or a local cache) the multimedia items, the Broker service selects and allocates various Transmoder services. Item Transmoder services receive from the Broker service a URI to the Download Manager service where they can obtain multimedia items using a simple web service call. Streaming Transmoder services, on the other hand, receive a reference to a TCP or UDP socket on which they can contact a Download Manager service to receive multimedia items in a streaming fashion. The Broker service also provides a reference to a Home Gateway service to all Transmoder services, so they know where they have to publish their results.

### 3.1. Extending Web Services for Streaming Multimedia

SOAP [34] has become the preferred standard for exchanging messages with web services. It is a simple one-way protocol that provides a flexible and extensible way to send structured and typed XML [35] data over a transport protocol. Sending various types and large loads of multimedia data with SOAP, however, can become complex and inelegant. A lot of extra work is involved when encoding the data (to fit the character encoding of the SOAP envelope) and splitting it into smaller chunks (to limit the effects of packet-loss in transport) [39].

That's where the Direct Internet Message Encapsulation (DIME) [18] comes into play. DIME offers a way to send (binary or text) attachments along with SOAP messages, regardless of their format and encoding. It is similar to MIME [11] but it can be parsed even more efficiently and it is conceived specifically for use with SOAP and web services. Using DIME, (multimedia) data can be sent together with its metadata in a SOAP message. Such a SOAP message can contain various information on the multimedia data (e.g. encoding, description and resolution). Providing metadata is commonly regarded as a key requirement for the successful processing and authoring of multimedia [15]. The ability to send metadata, in a separate description, along with multimedia in an open and standardized way is much more elegant and safe than sending that information on a custom basis.

We use DIME for two purposes: to send SOAP messages with multimedia attachments over HTTP and to stream large sets of multimedia data, together with their SOAP-packaged metadata, over TCP and UDP.

When streaming large amounts of multimedia data with DIME, the standard web service mechanisms are not sufficient. The life-time of a web service is limited to the duration of its web method call, while the streaming itself may take a long time. It may even take several hours in the case of real-time video streaming. Therefore, we introduce threads that run in the memory space of web services and handle the streaming and transmoding of the multimedia. Such a thread typically waits for another component to connect to its specific socket, and communicates using SOAP messages over a DIME stream on that port.

### 3.2. Broker Service

A Broker service selects and contacts Download Manager and Transmoder services upon a client's request. The selection it makes is derived from a track history it has stored in an internal database. A Broker decides upon the selection of specific services by analyzing a client's capabilities. This analysis results in a working space for Transmoder services, determined by the capabilities as indicated by a client. In our architecture, clients use a CC/PP-based description to advertise their capabilities [33]. Using this description, a Broker determines which descriptions and formats can be consumed by the client. It instantiates the appropriate services to provide the multimedia in these descriptions and formats.

Once the services are started and the client can start consuming the multimedia, the Broker service becomes available for other requests.

### 3.3. Home Gateway

The Home Gateway is a device that is placed, often by an Internet Service Provider (ISP), in the consumer's home. It acts as a gateway between the consumer's (wireless) Local Area Network (LAN) and the ISP. As such, it may play a key role in offering a guaranteed service when the consumer accesses information and multimedia on the Internet. Multimedia items form a significant portion of the data that's placed on the Home Gateway, which caches frequently used data and offers a maximum quality-of-service for multimedia consumption.

In our architecture, the Home Gateway is used to store and cache transmoded multimedia presentations so they can be consumed locally. When the consumer's environment (e.g. available bandwidth or processing power) changes during a presentation session, the Home Gateway may choose from a number of strategies to adapt the properties of the multimedia presentation:

1. If the changes remain within a given predictable range, scalable multimedia formats are likely to be able to handle the changes.
2. A multimedia item that was delivered by an Item Transmoder can be re-requested in a different form from the Item Transmoder that delivered it, using the

reference that the Item Transmoder gave when delivering the item, as explained in Section 3.5.

3a. When a Streaming Transmoder service is providing (a part of) the multimedia presentation, the bi-directional DIME channel that exists between the Home Gateway and the Streaming Transmoder service can be used to indicate the changes that occurred, enabling the Streaming Transmoder service to adapt the multimedia stream it delivers.

3b. The Broker service anticipates the range of changes that the Streaming Transmoder can handle. When this scope is exceeded, the Home Gateway may contact a new Broker service specifically for the new specification requirements of this multimedia item.

### 3.4. Download Manager Service

Our Download Manager services reside at the proxy-server level and provide for an abstraction of the protocols used by content creators. They offer the multimedia, which they downloaded over any known protocol, in a uniform way through web services attachments or a continuous DIME-stream. Upon request of a Broker service, they may start pre-fetching multimedia data from a content provider even before a Transmoder service requests it, caching the data in local memory. A Download Manager service may cache several versions of a multimedia item (as in simulcast), to facilitate the tailoring by Broker and Transmoder services.

### 3.5. Transmoder Service

Transmoder services form the distributed engine of our architecture. They are responsible for a range of transmoding tasks, yet each Transmoder service may support a limited transmoding functionality. Such tasks can be fairly simple, e.g. re-scaling a PNG image from one resolution to another. They can also be quite complex, e.g. extracting key frames from a video stream and sending them as a series of PNG images for consumer devices that do not support streaming video. Transmoder services can be very specialized, focusing on the tasks that they perform best without having to offer other transmoding functionality.

The services place the information they produce directly on a Home Gateway, which was referenced by the instructing Broker service. Apart from the multimedia item and metadata, this information also contains a reference to the generating Transmoder service itself, so the Home Gateway can access it at a later moment when a slightly different version of the multimedia item is required.

We introduce two families of Transmoder services:

- Item Transmoders receive requests of a Broker service to transmode single multimedia items. They retrieve and provide the items they handle using DIME over HTTP, as SOAP messages with attached multimedia data.

- Streaming Transmoders connect to given Download Manager services and Home Gateways using DIME over a streaming protocol. They continue to run as a thread for as long as the multimedia data continues to stream to the consumer. Streaming Transmoders can become quite complex, as multimedia streams have to be retrieved, transmoded and provided at an appropriate pace. It is very important to select good load balancing and buffering schemes, as they greatly influence the resulting multimedia presentation [21].

## 4. Case Study: Generating SMIL Presentations

To illustrate the added value of our architecture, we applied it to the following scenario. A Content creator has placed a small XML document that describes a simple presentation on a server. The presentation consists of some TIFF bitmap images that have to be displayed for a given duration, together with a textual description and an Ogg Vorbis sound clip [19].

A consumer wishes to view this presentation on a personal computer using the media-player *X-Smiles* [20]. Through its Home Gateway, it contacts a Broker service and provides the URI of the presentation and a CC/PP description of it's environment (player type and version, supported formats, screen resolution, etc.). The Broker decides upon the application of specific Transmoder Services. In this scenario a XML-to-SMIL2 transmoder (performing XSL transformations [28]) and a TIFF-to-PNG transmoder will be applied, since X-Smiles requires a SMIL-description of the presentation and it does not support TIFF bitmap images [31]. Both the sound clips and textual descriptions can be placed on the Home Gateway without transmoding if their required bandwidth and encoding are suitable for the consumer's environment.

The Broker returns the web service call and indicates at what location on the Home Gateway the appropriate presentation can be found. This presentation has been adapted as well as possible to the consumer's environment. Since a Broker service is only responsible for the presentation at initialization time, it may indicate that a presentation is ready for consumption even before all multimedia items have been placed on the Home Gateway, assuming that these items will be present when needed. The Home Gateway is responsible for re-requesting multimedia items when they are not present in the correct format during a presentation session.

We have also applied this scenario to an Apple QuickTime player, which offers no support for SMIL 2 and Ogg Vorbis [1]. Therefore, the Broker service applies a XML-to-SMIL1 transmoder and OGG-to-MP3 transmoder to the presentation and sound clips respectively, since these formats are supported by the player [27].

### 4.1. Experimental Results

The experiments we performed, show that our web service-oriented architecture is applicable for re-authoring multimedia presentations. Under ideal network conditions and server availability, our entire presentation was ready for consumption on the Home Gateway in a fraction of a second. The minor overhead caused forms a small inconvenience compared to the important merits of our architecture. Less than ideal network conditions may influence this time considerably, but such an extra delay is inherent to the transport of multimedia data and has no direct relationship to our architecture. The transport from content creator to consumer naturally would also have occurred without our architecture at the proxy server. Since our Download Manager service maintains several multimedia items in its cache, consecutive runs of the experiments show a significant decrease (up to 50%) of this total preparation time.

By introducing only a few image-to-image transcoding and XSLT services, our architecture is already able to transform a multimedia presentation from one format to another, taking into account the client's capabilities such as screen resolution, available bandwidth and supported SMIL version.

Figure 3 shows a screenshot of the X-Smiles player, playing a SMIL presentation that is generated by our architecture. In this version, the selected TIFF-to-PNG Transmoder service does not support image rescaling. Images that are too large for the player's display region are shown partially only. The textual description of each image is printed in the white text-box underneath the image-region.

The same presentation is shown in Figure 4, but here we selected a TIFF-to-PNG Transmoder service that does support rescaling of images. The black rectangle was drawn on the figure after taking the screenshot, to indicate the portion of the image that was visible in Figure 3.

Naturally, rescaling of the images can occur at the client, but that results in a significant overhead in the amount of data that has to be sent to the client. In our experiments, the image of Figure 4 has a size of 13 kB, while the image of Figure 3 is about 64kb. If the image would be rescaled at the client, this would result in a bandwidth overhead of $\frac{64-13}{13} = 392\%$.

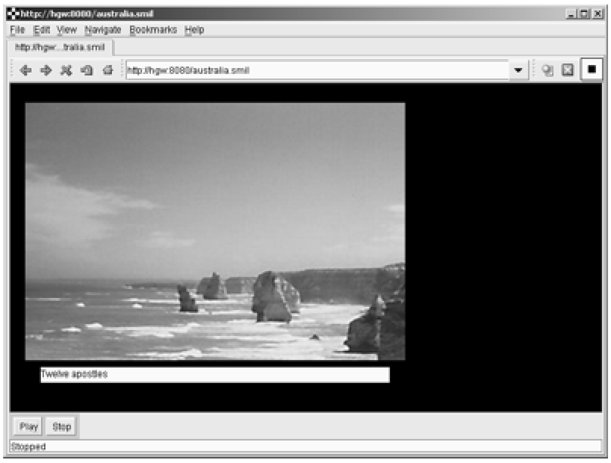


**Figure 3. The generated SMIL presentation without adaptation of the images' resolution. Large images do not fit in the image region and require a surplus of bandwidth.**

## 5. State of the Art and Related Work

Many approaches have been proposed for multimedia resource adaptation and capability negotiations, where terminals along with their request convey their preferences and capabilities. Standards have been developed by standard organizations such as W3C, WAP Forum, 3GPP and ISO. W3C and IETF focus on facilitating server-side decision making on the mechanisms of content adaptation and content delivery. These mechanisms rely on user information presented by the HTTP/1.1 content negotiation capability and the composite capabilities/preferences profile (CC/PP) framework developed by W3C [33]. CC/PP is based on the Resource Description Framework, a general purpose metadata description language also developed by W3C [36].

There are two popular techniques for sending raw (e.g. binary) data with XML documents: by inclusion and by reference. Data can be included in XML documents by encoding it to a sequence of base64 or hexadecimal characters. This way, none of the data can wrongfully be interpreted as a part of the XML document. However, our experiments have shown that these encoding approaches can bring about a significant overhead, up to 200%, to the XML documents. Data can also be sent separately, referenced from within the XML document. Both WS-Attachments and SOAP with Attachments (SwA) follow this approach, employing DIME and MIME respectively. They both introduce a separate data model for the XML message and the raw data. As a consequence, the creation and processing of these messages becomes more complex. Furthermore, applying security and routing mechanisms to DIME'd or

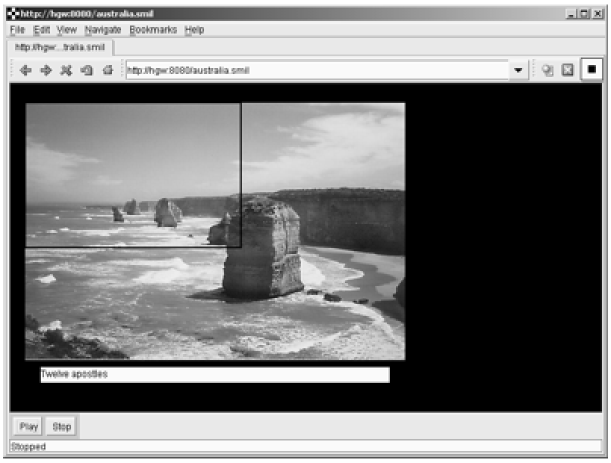


**Figure 4. The generated SMIL presentation with adapted resolution of the images. The black rectangle shows the area that would have been visible without resolution adaptation.**

MIME'd XML documents can be a difficult and error-prone task. Even though our approach could employ any of the described techniques, in these experiments we have chosen to use WS-Attachments and DIME.

Several architectures have been proposed for multimedia re-authoring. Digestor [5] focuses on re-authoring of WWW pages. Bellavista [4] describes an active middleware to control quality-of-service, specifically for streaming Video-On-Demand. Jia Zhang proposes a SOAP-oriented framework to support device-independent multimedia web services [39]. The framework introduces a mechanism for transporting large multimedia streams from and to web services, which offers an alternative to DIME. A high-level system architecture is proposed by Roy [21]. It focuses on load balancing and resource distribution without providing details on the standards and mechanisms that are used to build the infrastructure. TranSquid [17] focuses on caching in the transmoding services, specifically for e-commerce environments. Ellen Zhang [38] describes a translation proxy for connecting different proprietary players and servers.

The MPEG-21 multimedia framework is an open framework for multimedia delivery and consumption for use by all the players in the delivery and consumption chain [12]. It supports the augmented use of multimedia resources across a wide range of networks and devices used by different communities [6]. Within MPEG-21, *Digital Item Adaptation* (DIA) [7] describes the manipulation of multimedia content in a networked context, to tailor for the needs of end-user terminals. Such manipulation can consist of the transcoding of a video clip, the translation of a text, etc. [9].

## 6. Conclusions and Future Work

A direct result of our approach is that content creators need no longer bother with a multitude of client platform specifications and connecting networks. Their only concerns are the multimedia applications they wish to publish, and the data formats in which those applications are stored internally. Naturally, the format in which the multimedia is offered to the network has a great influence on the multimedia experience by the consumer. However, a content creator's server is no longer aware of every single scenario in which clients may want to consume its multimedia data. As such, the server will be less vulnerable to obsolescence when new client platforms arise, since multimedia applications can be adopted to new emerging consumer environments by the service in the network.

Our architecture is based entirely on recent open standards, making it open and flexible. It supports complex multimedia presentations, composed of any combination of multimedia items, like images, text, video, audio, etc.

Recent extensions to web services, like encryption and routing, have opened a whole new world of service-oriented applications. We will investigate their applicability to and added-value for multimedia re-authoring architectures.

Furthermore, we will investigate caching algorithms that can be applied in order to enable efficient support of large numbers of clients.

## 7. Acknowledgments

This work is a result of a joint collaboration between Vlaamse Radio en Televisie (VRT, public broadcaster of Flanders, Belgium), the Interuniversity MicroElectronics Center (IMEC), Universiteit Gent (UGent) and Vrije Universiteit Brussel (VUB). This is one of the E-VRT projects funded by the Flemish government.

## References

[1] Apple inc. quicktime. http://www.apple.com/quicktime.

[2] Microsoft inc. windows media technology. http://msdn.microsoft.com.

[3] Realnetworks inc. realsystem streaming platform. http://www.realnetworks.com.

[4] P. Bellavista, A. Corradi, R. Montanari, and C. Stefanelli. An active middleware to control qos level of multimedia services. In *Proceedings of the eighth IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS)*, pages 126–132, 2001.

[5] T. Bickmore and B. Schilit. Digestor: Device-independent access to the world wide web. In *Proceedings of the sixth World Wide Web Conference*, pages 655–663, 1997.

[6] J. Bormans. *MPEG-21 Requirements 1.2 – ISO/IEC JTC1/SC29/WG11/N4988*, July 2002.

[7] J. Bormans, J. Gelissen, and A. Perkis. MPEG-21: The 21st century multimedia framework. *IEEE Signal Processing Magazine*, pages 53–62, Mar. 2003.

[8] C. Brooks, M. Mazer, S. Meeks, and J. Miller. Application-specific proxy servers as http stream transducers. In *Proceedings of the fourth World Wide Web Conference*, 1995.

[9] R. De Sutter, S. Lerouge, J. Bekaert, and R. Van de Walle. Dynamic adaptation of streaming mpeg-4 video for mobile applications. In *Proceedings of the Euromedia Conference*, pages 185 – 190, Apr. 2003.

[10] A. Fox and E. Brewer. Reducing www latency and bandwidth requirements by real-time distillation. In *Proceedings of the fifth World Wide Web Conference*, 1996.

[11] N. Freed and N. Borenstein. *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, Nov. 1996. rfc 2045.

[12] ISO/IEC. *Information technology – Multimedia framework (MPEG-21) – ISO/IEC 21000*.

[13] ISO/IEC. *Information technology – Generic Coding of Moving Pictures and Associated Audio (MPEG-2) – ISO/IEC 13818*, 1995.

[14] ISO/IEC. *Information technology – Coding of audio-visual objects (MPEG-4) – ISO/IEC 14496*, 1998.

[15] ISO/IEC. *Information technology – Multimedia content description interface (MPEG-7) – ISO/IEC 15938*, 1999.

[16] ITU-R. *BT.804: Characteristics of TV receivers essential for frequency planning with PAL/SECAM/NTSC television systems*. http://www.itu.int.

[17] A. Maheshwari, A. Sharma, K. Ramamritham, and P. Shenoy. Transquid: Transcoding and caching proxy for heterogeneous e-commerce environments. In *Proceedings of the 12th International Workshop on Research Issues in Data Engineering (RIDE)*, 2002.

[18] H. F. Nielsen, H. Sanders, R. Butek, and S. Nash. *Internet Draft: Direct Internet Message Encapsulation (DIME)*. IETF, 2002. http://www.ietf.org/internet-drafts/draft-nielsen-dime-02.txt.

[19] Ogg Vorbis Team. *Ogg Vorbis comment field specification. Web notes from the developers*, Feb. 2001. http://www.xiph.org/ogg/vorbis/doc/v-comment.html.

[20] K. Pihkala, N. von Knorring, and P. Vuorimaa. Smil in x-smiles. In *Proceedings of the Seventh International Conference on Distributed Multimedia Systems*, 2001.

[21] S. Roy, M. Covell, J. Ankcorn, S. Wee, and T. Yoshimura. A system architecture for managing mobile streaming media services. In *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 408–413, 2003.

[22] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. *RTP: A transport protocol for real-time applications*, Jan. 1996. rfc 1889.

[23] S. Van Assche, F. Hendrickx, and L. Nachtergaele. Multi-channel publication using MPEG-21 DIDL and extensions. In *Proceedings of the Twelfth International World Wide Web Conference*, 2003. Poster.

[24] S. Van Assche, F. Hendrickx, N. Oorts, and L. Nachtergaele. Multi-channel publishing of interactive multimedia presentations. *Elsevier Computer and Graphics*, pages 193–206, Apr. 2004.

[25] M. Van Der Schaar, Q. Li, and L. Boland. Internet video streaming with FGS. In *Proceedings of the 7th International Conference on Information Systems, Analysis and Synthesis (ISAS)*, 2001.

[26] W3C. *Cascading Style Sheets, level 2 – W3C Recommendation*, May 1998. http://www.w3.org/TR/1998/REC-CSS2-19980512.

[27] W3C. *Synchronized Multimedia Integration Language (SMIL) 1.0 – W3C Recommendation*, June 1998. http://www.w3.org/TR/1998/REC-smil-19980615.

[28] W3C. *XSL Transformations (XSLT) 1.0 – W3C Recommendation*, Nov. 1999. http://www.w3.org/TR/1999/REC-xslt-19991116.

[29] W3C. *Extensible HyperText Markup Language (XHTML) 1.0 – W3C Recommendation*, Jan. 2000. http://www.w3.org/TR/2000/REC-xhtml1-20000126.

[30] W3C. *Scalable Vector Graphics (SVG) 1.0 – W3C Recommendation*, Sept. 2001. http://www.w3.org/TR/2001/REC-SVG-20010904.

[31] W3C. *Synchronized Multimedia Integration Language (SMIL) 2.0 – W3C Recommendation*, Aug. 2001. http://www.w3.org/TR/2001/REC-smil20-20010807.

[32] W3C. *Web Services Description Language 1.1 – W3C Recommendation*, Mar. 2001. http://www.w3.org/TR/wsdl.

[33] W3C. *Composite Capability/Preference Profile – W3C Recommendation*, Mar. 2003. Working Draft, http://www.w3.org/TR/CCPP-struct-vocab.

[34] W3C. *Simple Object Access Protocol (SOAP) 1.2 – W3C Recommendation*, June 2003. http://www.w3.org/TR/2003/REC-soap12-part1-20030624/.

[35] W3C. *Extensible Markup Language (XML) 1.0 – W3C Recommendation*, Feb. 2004. http://www.w3.org/TR/2004/REC-xml-20040204.

[36] W3C. *RDF Vocabulary Description Language 1.0: RDF Schema – W3C Recommendation*, Feb. 2004. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/.

[37] D. Xu, D. Wichadakul, and K. Nahrstedt. Multimedia service configuration and reservation in heterogeneous environments. In *Proceedings of the 20th International Conference on Distributed Computing Systems (ICDCS)*, 2000.

[38] E. Zhang, D. Towsley, and J. Wileden. Towards interoperable multimedia streaming systems. In *Proceedings of the 12th International Packetvideo Workshop (PV)*, 2002.

[39] J. Zhang and J.-Y. Chung. A SOAP-oriented component-based framework supporting device-independent multimedia web services. In *Proceedings of the IEEE Fourth International Symposium on Multimedia Software Engineering (MSE)*, 2002.