

A Context-Aware Architecture for QoS and Transcoding Management of Multimedia Streams in Smart Homes

Raffaele Bolla, Matteo Repetto
Dept. of Communication, Computer and
System Science (DIST) – University of
Genoa
Via all'Opera Pia 13, 16145 Genova, Italy
{raffaele.bolla, matteo.repetto}@unige.it

Saar De Zutter, Rik Van de Walle
Dept. of Electronics and Information
Systems, Multimedia Lab, Ghent University
and Interdisciplinary Institute for Broadband
Technology (IBBT)
Gaston Crommenlaan 8 bus 201, B-9050
Ledeberg-Ghent, Belgium
{saar.dezutter, rik.vandewalle}@ugent.be

Stefano Chessa, Francesco Furfari
ISTI – CNR
via Moruzzi 1, 56100 Pisa, Italy
{chessa, furfari}@isti.cnr.it

Bernhard Reiterer, Hermann Hellwagner
Dept. of Information Technology (ITEC) –
Klagenfurt University
Universitätsstraße 65–67, A–9020 Klagenfurt,
Austria
{reiterer, hellwagn}@itec.uni-klu.ac.at

Mark Asbach, Mathias Wien
Institute of Communications Engineering
(IENT) – RWTH, Aachen University
D-52056 Aachen, Germany
{asbach, wien}@ient.rwth-aachen.de

Abstract

Current trends in smart homes suggest that several multimedia services will soon converge towards common standards and platforms. However this rapid evolution gives rise to several issues related to the management of a large number of multimedia streams in the home communication infrastructure. An issue of particular relevance is how a context acquisition system can be used to support the management of such a large number of streams with respect to the Quality of Service (QoS), to their adaptation to the available bandwidth or to the capacity of the involved devices, and to their migration and adaptation driven by the users' needs that are implicitly or explicitly notified to the system. Under this scenario this paper describes the experience of the INTERMEDIA project in the exploitation of context information to support QoS, migration, and adaptation of multimedia streams.

1. Introduction

Current trends in smart homes suggest that several multimedia services will soon converge towards common standards and platforms [1]. Such convergence will enable the exchange of several multimedia streams between a number of personal communication devices and home infrastructural

devices, such as residential gateways or multimedia centers with advanced multimedia interfaces.

However this rapid evolution gives rise to several issues related to the management of a large number of streams in the home communication infrastructure. A critical aspect in this scenario is how to make this convergence as automatic as possible, in order to avoid any manual user intervention. To this aim, an issue of particular relevance is how a context acquisition system can be used to support the management of such a large number of streams with respect to the Quality of Service (QoS), to their adaptation to the available bandwidth or to the capacity of the involved devices, and to their migration and adaptation driven by the users' needs that are implicitly or explicitly notified to the system. Consider for example, the case of a video transcoder able to dynamically adjust the output video quality (and thus the required bandwidth) that has to send a video stream to a multimedia renderer through the network. Such a transcoder may negotiate resource allocation with a QoS management service and, as a consequence, it may dynamically adjust the quality of its stream in order to adapt it to the available bandwidth. In this simple scenario it is possible to identify two components of the QoS management system: a component that acts as a network monitor in order to identify the activation of new communication streams and implicitly determines their needs, and a component that implements the QoS policies. The

latter acquire the information produced by the monitoring components and negotiates with the applications about the resource allocation. Moreover, the QoS policies may also take into account other context information, such as the position of the users, their activities, and their preferences (for example, to determine a priority between the different applications). In other words, the QoS management system can be seen as a context-aware system that aggregates context information from a large number of context sources including network monitors, environmental sensors, users' profiles, etc.

Under this scenario, this paper describes the experience of the INTERMEDIA project [2] in the exploitation of context information to support QoS, migration, and adaptation of multimedia streams. More specifically, we propose an architecture in which a context service provides information about the user, the environment, and the network status to a QoS service using a simple and high-level interaction based on the UPnP standard [3]. Using this context information we describe how the multimedia content can be dynamically adapted and possibly migrated according to the network and users' condition.

The rest of the paper is organized as follows. Section 2 presents basic concepts and services to support smart access for multimedia contents and Section 3 introduces the INTERMEDIA communication infrastructure. The QoS management services and the integration layer called SAIL that enables integration of the different components of the INTERMEDIA architecture into a UPnP-based peer-to-peer (P2P) network are presented in Sections 4 and 5, respectively. Sections 6 and 7 discuss the metadata generation and the content adaptation servers. Conclusions are drawn in Section 8.

2. Basic Services for Smart Multimedia Content Access

The reference scenario depicted in Figure 1 can effectively represent a smart environment where users wish to access several kinds of multimedia contents available from different sources (television broadcasting, video-on-demand, IP telephony, etc.). These sources can be accessed by means of appropriate devices and/or applications inside the home; they are referred to as *Content Source i* in Figure 1. Contents are “played” by means of renderers (indicated as *Media Renderer i*), which are scattered around the house. Different renderers have different multimedia capabilities, mainly in terms of audio/video/textual formats and supported codecs; the *Content Adapter* elements are able to adapt media from its original format to the specific renderer capabilities. In such a scenario we are interested in two main issues:

- automatic adaptation of media streams;
- efficient usage of network resources.

These two issues are somehow related: video and audio coding needs adequate support from the network in order to deliver packets within the required latency and with minimal losses. A QoS service is therefore needed to accomplish this task. Without any other element, users may have to manually change the renderer and the transmission coding when media quality degrades due to network congestion. This is avoided in the INTERMEDIA architecture by modelling the QoS Manager as a context-aware system, which collects data from a context acquisition system (depicted as circles and rhombuses in Figure 1).

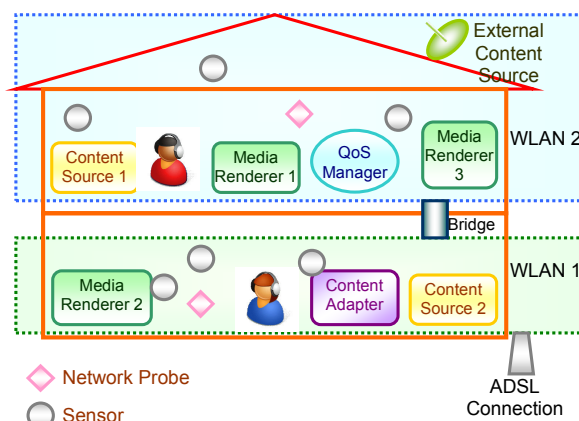


Figure 1. The INTERMEDIA scenario.

The rest of this section gives a brief overview of the main functional elements identified in the smart home environment: context acquisition systems, QoS issues in the home environment, and transcoding.

2.1. Context-aware Systems

The first efforts to introduce context-awareness have been related to the localization of users [4]. Localization is still one of the main building blocks of context-aware systems, although recently the concept of context-awareness has been enriched to take into account more general environmental parameters, where the meaning of the term “environmental” is as broad as possible. Environmental parameters may refer to user physiological/emotional data, user actions/movements, user identity, status of the surrounding environment, location, time, profiles, agendas and data referable to the user, and even presence and context of other users [5][6]. Context-aware systems are particularly useful to support mobile applications since the context may change rapidly with mobility (of the user and/or of the environment), and the system should react rapidly to such context changes.

In general, the architecture of a context-aware system includes the following layers: sensors, raw data retrieval, pre-processing, storage/management, and application.

The sensor layer includes not only the hardware sensors (physical sensors) but also any data source providing context information, for instance virtual sensors which offer data available from applications or services (e.g., data deducted by a specific use of a browser by the user) and logical sensors which combine information obtained from physical and virtual sensors (e.g., the location of the user associated to an action on a browser). Recent research issues focused on networking between physical sensorial devices, to build so called Wireless Sensor Networks (WSN [7]).

Context-aware systems can be implemented according to different architectures, which in [8] are classified according to the way the contextual information is collected, in terms of *direct sensor access*, *middleware infrastructures*, or *context servers*.

2.2. Quality of Service in Home Environments

Quality of Service can be managed at different layers in the network stack. End-to-end guarantees can be effectively supplied by layers with an end-to-end scope; the typical solution is to implement QoS at the network layer with two different approaches, namely Integrated Services (IntServ) [9] and Differentiated Services (DiffServ) [10]. IntServ is a fine-grained, flow-based architecture for service guarantees on the Internet. Signalling is done using the RSVP protocol [11], which exchanges information about what the application requests from the network and what the network can guarantee. DiffServ is a coarse-grained, class-based architecture for service differentiation on the Internet. The network defines different classes of traffic, with different QoS requirements. In complex networks, the combination of IntServ at the network edge and DiffServ in the network core can offer the features of IntServ together with the ease of scalability and implementation of DiffServ.

Dealing with QoS at the network layer enables to effectively distinguish traffic flows (for both hosts and applications), to use the same mechanisms across heterogeneous networks and to offer a uniform view of QoS to users and applications. However, the end-to-end policies must be translated into specific technology-dependent mechanisms in the traversed networks. Currently, QoS is defined by IEEE 802.1p for Ethernet switches, in terms of queuing disciplines. For Wi-Fi networks, IEEE 802.11e defines the same priorities, which correspond to a limited number of Access Categories (AC); each AC is associated to a different queue and different operational parameters, in order to get several levels of priorities among packets both inside a single device and among different ones.

Applications usually do not have direct control on data link operations: instead, there are standard mechanisms to map L3 (i.e., network) policies into

specific L2 (i.e., link layer) mechanisms. For example, the 802.11e priorities are derived automatically from the IP DSCP codepoints [12]; applications that wish a specific treatment for their traffic at the data link layer have to mark their IP packets accordingly. A QoS server is typically needed to manage the priority level of each traffic stream in order to achieve system efficiency and fairness.

2.3. Exploiting Metadata for Adaptation

Because of the large variety of possible media items, content delivery heavily benefits from any available knowledge about how the content is to be adapted. Regarding the application scenario, the adaptation decisions generally target maximum visual quality at the given bandwidth. However, since there are multiple dimensions of adaptation, the optimum decision requires taking both the context of the user and the characteristics of the media into account. Metadata represents this knowledge about the media.

Sometimes metadata is stored within the media (like EXIF tags in digital photos), in other cases it might be available from other locations (like electronic programme guides for TV). It might also be possible that helpful information is implicitly contained in the coded multimedia data itself, but has to be extracted. The detection of shot boundaries and spatial segmentation are typical examples of such cases [13]. In INTERMEDIA, an extensible set of tools has been defined that allows annotating multimedia data whenever new media items become available.

We can roughly make a distinction between low-level (signal level) annotations, medium-level (structural level) annotations, and high-level (semantic level) annotations. Annotation can be performed either by humans or by machines; semi-automatic hybrid procedures with varying degrees of automation are also possible.

Higher-level annotations typically allow for more effective adaptation decisions. It might, for example, be good to reach a given bandwidth bound by reducing temporal resolution for scenes with low motion activity (structural level), but it might be even better to utilize spatial or SNR scalability in image regions that are not important to the human viewer (requiring semantic level information like region of interest (ROI) information) [14].

3. An Architecture inside INTERMEDIA

The INTERMEDIA vision goes beyond the traditional communication paradigms, where the user usually has to adapt to the available communication system. INTERMEDIA has adopted a user-centric approach, where the communication facilities are built around the user: dynamic networking, context acquisition, content sharing and adaptation, security

and DRM. The INTERMEDIA vision and scenarios are summarized on the project Web site [2]. Context information is required to identify the user and its representation (profiles, requirements); a number of heterogeneous interfaces on different kinds of devices are available to the user and content adaptation is required in order to adapt multimedia sources to such interfaces.

Multimedia transmissions usually consume a lot of bandwidth, and this kind of traffic is sensitive to delay and jitter. Our architecture is devoted to deal with such kind of traffic in an efficient and scalable way, taking into account the fact that home networks are often made of simple and cheap devices, unable to support sophisticated QoS control mechanisms.

Apart from the obvious presence of a wireless network, we can identify three main components in our architecture:

- the **multimedia subsystem**, composed of content sources, multimedia renderers and content adapters;
- the **QoS management subsystem**, which is responsible for the communication management by taking into account as much context information as possible;
- the **context acquisition subsystem**, which gathers context information and exports it to the whole system.

The **multimedia subsystem** includes all the elements able to provide, manipulate or use multimedia content, i.e., sources, renderers and adapters such as transcoders. Several multimedia content sources can be taken into account in a smart home environment; these sources can be external, for example television channels on the satellite, DVB-T, or the Internet, or they can be internal, in form of multimedia data stored in PCs, PDAs, mobile phones, or other multimedia appliances available in the home. In the same way, a number of renderers might be used to play the content: these renderers can be static (such as televisions, stereos, PCs) or mobile. In the latter case, the renderers are typically mobile phones or PDAs or other wearable devices. As there is potentially a large number of heterogeneous renderers, with different capabilities, the architecture takes into account also content adapters, which are able to adapt the format of the available multimedia content to the actual media renderer in use. These tools include video transcoders that are able to dynamically change the video (audio) stream characteristics, e.g., in terms of used bandwidth. The multimedia subsystem also contains a controller component that governs the behaviour of the multimedia components, especially the interaction between them, based upon information gathered from the other subsystems. Currently, the controller is realized as a UPnP control point, exceeding the scope of usual UPnP AV control points by offering the

capability to redirect content transport via adaptation components, and to transfer running multimedia sessions from one renderer to another.

The **QoS management subsystem** acts as an admission controller and streaming manager; it is a context-aware subsystem, which gathers information from a context acquisition subsystem. The QoS Manager exports information to the multimedia applications (e.g., multimedia subsystem elements) about network usage and available resources, so that the latter can adapt their coding as appropriate; it might directly suggest applications to reduce or increase their bandwidth and, if supported by the physical and MAC layers, it manages requests of bandwidth allocation.

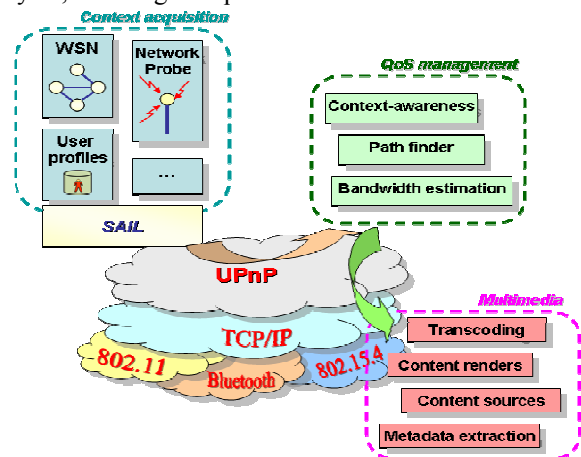


Figure 2. An architecture for smart homes.

Finally, the **context acquisition subsystem** represents one of the most important elements in a user-centric framework. Context information enables multimedia applications and the QoS Manager (as well as all other kind of managers) to become aware of environmental conditions and users' preferences without requiring any manual intervention from users. In the specific scenario, a wireless sensor network monitors the environment and the users; it provides information related to user localization, to users' activities, to their use of appliances, devices or other equipment, and to general conditions of the environment (light, temperature, humidity, etc.). Moreover, the network is monitored by a set of "probes" that provide information about the current active communication flows.

Figure 2 shows how the three components are integrated in the proposed architecture. Apart from the obvious presence of the TCP/IP stack, this architecture builds upon well known wireless standards including IEEE 802.11 for multimedia communications and management, Bluetooth for interconnection with personal devices, and IEEE 802.15.4 for the WSN. Multimedia communications rely on control and transport protocols, such as RTP, RTSP, and SIP; in addition, the UPnP technology helps in making the

different devices usable out-of-the-box, thus unburdening the user of tedious manual configuration. The architecture exploits UPnP to construct a P2P network of the components described above; in particular, the architecture exploits the device and service discovery mechanisms of UPnP.

The context acquisition subsystem gathers very heterogeneous information; in the proposed architecture, a common interface to access and interact with the various components is provided by a software layer called SAIL (Sensor Abstraction and Integration Layer), which embeds each component and provides an abstraction of the component that is exported in terms of a UPnP device. A description of SAIL is given in Section 5.

4. The QoS Manager Services

The QoS Manager is hosted in a server that offers the QoS services to the applications and that acquires context information from the WSN and from the Network Probes. The interaction among these components happens by means of UPnP; the QoS Manager is a component of the QoS UPnP architecture.

The QoS Manager should provide information about estimated network traffic conditions of both single subnetworks (e.g., one WLAN) and paths crossing multiple subnetworks; it should also take care of resource reservation if network devices support such a feature.

The monitoring service provides the QoS Manager with a quite varied set of information (mean and instantaneous bandwidth, number of active flows, arrival and service rates, etc.). This information is updated dynamically according to a configurable period and is a direct product of the Network Probes; it is merged with context information about user/renderer localization in order to implement a real context-aware system. The actual design of the merging process is out of the scope of this paper. The Network Probe does not require any specific QoS features on all intermediate devices; it should be introduced in the QoS architecture profile of UPnP.

When the application requires information on a path between different subnetworks (namely, when the source and the destination are not in the same physical network), the service combines the information on the status of the different subnetworks crossed by the path connecting the two hosts, and the user's location and statistics.

The service described above only provides feedbacks and "suggestions" to QoS-aware INTERMEDIA applications, but in absence of low-level mechanisms at the MAC layer of the networks it cannot force any QoS policy; it can only implement a simple *Call Admission Control* function. However, if the network and MAC layers provide QoS mechanisms

(for example, if IEEE 802.11e is used), the QoS Manager can fully act as a Bandwidth Broker. In such a case, when requests for new communication streams with given characteristics (bandwidth, latency, burstiness) are received, the QoS Manager can attribute to each request a priority depending on the user and on the applications involved in the communication and, based on the information about the available network resources, it can also grant network resources (e.g., bandwidth, queue space). Such interaction is described in the UPnP QoS profile [3]. In any case, the admission of a new flow may require a renegotiation of resources currently used by other applications, if such applications can dynamically adjust their requirements. This should avoid network congestion and consequently prevent applications from experiencing any quality degradation due to a drop of resources.

5. The Context System

This section presents the context system by which information about users and networks is collected. It consists of a common layer responsible for the sensors' abstraction and integration (called SAIL); virtual sensors (WSN and network measures) are modelled and exported by this layer.

5.1. The Virtual Sensor Model

A Sensor Node models an abstract sensor embedding a set of abstract transducers, actuators, and services (that are used to model complex services, such as those offered by ZigBee). The concept of Sensor Node is not bound to any specific hardware or software platform for the WSN, furthermore a Sensor Node may or may not correspond to a physical sensor of the WSN. In fact, this abstraction can be used to model a single physical sensor, a group of physical sensors, or a virtual sensor, such as a Network Probe. For example, two different physical transducers in a sensor can be modelled by a single Sensor Node with a single abstract transducer.

The services can be accessed by send and receive primitives, while transducers and actuators can be accessed according to push or pull models. In the push model, the Sensor Node automatically sends the data to the upper layers, while in the pull model the data has to be explicitly requested by the upper layers.

5.2. The SAIL Architecture

SAIL is organized in three layers, namely the Access, Abstraction, and Integration Layers, constructed over an OSGi framework [15] and shown in Figure 5. The OSGi specification defines a service-oriented, component-based environment for developers and offers standardized ways to manage the software life cycle for Java applications.

The SAIL Access Layer defines a minimal set of functionalities that any WSN application should provide, either on its own or by means of an application adapter. This layer interacts directly with the WSNs to implement the Sensor Node. To this end, it comprises a set of components called Sensor Application Drivers (SADs), each of which communicates with an application running on a WSN. The SAD exports the WSN functionalities in terms of a Service Provider Interface (SPI) [17] that is specified by the Abstraction Layer.

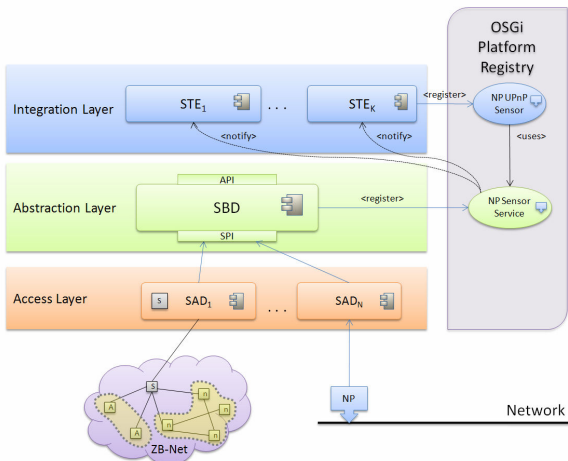


Figure 3. The SAIL architecture layers and the OSGi platform.

The Abstraction Layer is implemented by a single component called Sensor Base Driver (SBD). The SBD defines the SPI interface that must be implemented by the SAD. The SPI is a general interface that is designed to support different WSN application models. The SBD can be thought of as a high-level driver which registers the SAD in the OSGi framework. The SBD also implements an API that is used by the Integration Layer.

The Integration Layer exports the OSGi services registered by the Abstraction Layer to client applications. To this purpose it encapsulates different exporters, called Sensor Technology Exporters (STEs), suitable to provide access to the OSGi services using different technologies. In the current implementation the Integration Layer comprises exporters for UPnP and PERSONA [16].

5.3. The Network Probe as a Virtual Sensor

Usually, network measures are only available from internal hardware and software monitoring of network equipment (bandwidth usage, queue sizes, queue latencies, and so on). Often such measurements can be realized only in hardware and are thus quite expensive to supply. This leads to the unavailability of suitable monitoring tools for most widespread low-end network access hardware, such as access points and Ethernet switches.

In the proposed architecture, we monitor the network by means of software probes, whose aim is to collect as much information as possible about the network traffic. Such tools can be hosted on general purpose PCs as well as on network equipment; they take advantage of the intrinsic shared nature of the wireless medium and the replication capabilities¹ of many Ethernet switches. Each Network Probe makes a flow-based classification at the application layer and keeps track of any flow above the transport layer. For each flow a number of parameters is kept: total number of packets/bytes seen so far, QoS class, average and current bandwidth, estimation of the application protocol from which the flow has been generated, etc.; other measurements can be retrieved from aggregated data, such as number of active and terminated flows, statistical properties of any given class of flows (mean and instantaneous arrival rate and flow duration), bandwidth for different QoS classes.

The classification engine is build upon the *Click Modular Router* [18] tool running in user-space, while specific modules have been developed in order to recognize the application protocol generating the flow (these modules are based on a patter-matching schema derived from L7-filter [19]), associate each packet with a specific flow, and update statistics accordingly.

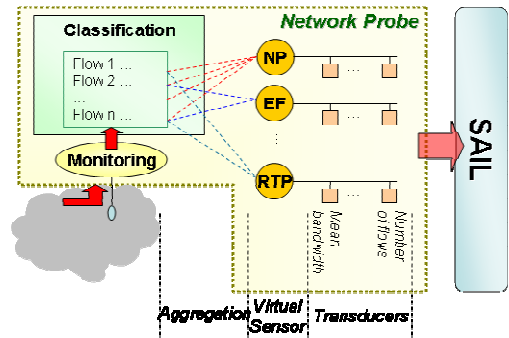


Figure 4. Architecture of the Network Probe.

The Network Probe can be viewed as a particular type of sensor, and thus it is virtualized within SAIL as a virtual sensor, where the transducers are associated to the measurements described above: mean and current bandwidth, number of active flows, mean and current arrival/departure rates, and number of bytes/packets arrived (see Figure 4). Such measures are available for different levels of aggregation, in particular we model each Network Probe (including network traffic below the transport layer), all the three main QoS classes [10], namely EF (Expedited Forwarding, [12]), AF (Assured Forwarding, [20]) and BE (Best Effort), and every recognizable application (HTTP, RTSP, RTP, DNS, etc.) to a different Virtual Sensor. This provides

¹ Some L2 switches can replicate all traffic from all or a subset of their interfaces to any given interface, in order to allow an external device to monitor all switched traffic.

the QoS Manager with a powerful set of information and the ability to distinguish among different classes of service, protocols and applications.

6. Metadata Generation for Context-aware Content Adaptation

The purpose of metadata extraction and generation is to allow for context-aware adaptation. Therefore, only those characteristics of the content have to be described that can be interrelated with both a user's context (including the networking conditions) and available adaptation techniques. In the INTERMEDIA scenario, we are mainly interested in metadata concerning temporal segmentation, Region of Interest (ROI) and scene classification.

Temporal segmentation. The general structure of audio as well as video material can be usually described as temporally concatenated pieces of media. Correspondingly, the first step to acquire information about multimedia content is to deduce this temporal structure (structural level) from hints found on the signal level. The INTERMEDIA scene cut detection tools [22], for example, use colour histograms to detect shot boundaries. A resulting initial description is subsequently stored in MPEG-7 format and can be accessed through the context server.

Region of Interest. In addition to the temporal domain, spatial information is also meaningful structural level information to be made available by the context server. Video object segmentation is a standard task in computer vision. Based on low-level characteristics like motion, colour, and texture, INTERMEDIA tools separate objects from background. This allows crop & scale approaches to adapt video to lower resolution screens (see [14]), but it also allows performing unequal SNR scalability, i.e. cutting down the bitrate for spatial parts with lower significance.

Scene classification. The INTERMEDIA video tools include scene classification into genres (e.g., anchor person of a news cast, weather forecast, or sports), which enables balancing multiple streams that share bandwidth. During scenes that match a preference of the user (via user context information), the system would acquire a larger slice of the bandwidth, while during scenes or genres that a given user does not value that much, the system may donate parts of a user's allowed bandwidth to users that currently stream scenes of *their* interest.

Useful domains of adaptation here are related to those expected from scalable media representations. Adaptation tools therefore allow changing (i.e., lowering) the data rate of encoded media by lowering its quality in terms of spatial, temporal, or SNR resolution. This might be possible by simply discarding

parts of hierarchically-encoded streams but may also require complex manipulations to the encoded data or even transcoding by re-encoding. To make the context-aware adaptation processes in Section 7 format-agnostic, high-level descriptions of the structure of the scalable content (i.e., compressed multimedia bitstreams) have to be generated and made available through the context server. This structural metadata can be transformed (an operation in the semantic domain) in order to reflect desired adaptations of the content, and can then be used to automatically create adapted versions of the multimedia streams. In INTERMEDIA, we describe the compressed bitstream structure in eXtensible Markup Language (XML), more specifically, we automatically generate MPEG-21 DIA generic Bitstream Descriptions (gBSDs) [21]. This shifts the complexity of the content adaptation process from the binary (compressed) domain to the XML domain.

7. Adaptation Servers

The adaptation process takes care of transforming a given content into the most suitable media format, according to the context information (namely renderers' capabilities, user preferences and network conditions). Adaptation could resolve into both media transcoding and content manipulation. The former is mostly related to transmission formats and codecs as controlled by a set of parameters, which include references to formats to be used as well as numeric properties for the resulting content, such as temporal frequency (frame rate, audio sampling rate), bitrates, frame size and height. The latter takes into account the content to be transferred and works at a higher layer by managing logical entities, such as scenes and objects.

The optimal settings of the transcoding parameters for the current situation must be determined by the adaptation decision taking engine (ADTE). The optimization problem to be solved by the ADTE can be specified by three components from MPEG-21 Digital Item Adaptation (DIA) [21]: Universal Constraint Descriptions (UCD) expresses fundamental constraints on variables that represent various media characteristics (e.g., bandwidth); Usage Environment Description (UED) characterizes terminal capabilities, network characteristics (which would be repeatedly updated using the aforementioned QoS Manager), user properties and relevant factors of the environment; and Adaptation Quality of Service (AQoS) descriptions enumerate possible values for adaptation parameters, resulting values (such as quality measures) and relationships between the values, e.g., only certain tuples are allowed for frame height and width, and such a tuple, along with other parameters, is input to a function that computes the resulting video's bitrate.

After a multimedia session is started with adaptation parameters computed according to the initial situation, the controller of the multimedia subsystem maintains the UED according to regular queries of the available context information, including results of network probing. If the available bandwidth is suddenly too small for the currently produced bitrate, the ADTE is invoked again, giving new parameters that are then used for adapting the rest of the media stream. Also, if the bandwidth rises by a significant amount, adaptation parameters will be updated by the ADTE.

Semantic and ROI-based content adaptations are two scenarios supported by the services for smart multimedia access and the INTERMEDIA architecture discussed in this paper. Semantic adaptation enables the extraction of desired content fragments by linking the high-level descriptions of the multimedia streams to the temporal segmentation metadata (see Section 6) and the user preferences (available through the context acquisition subsystem). In smart home environments, scene classification (see Section 6) can be used together with semantic adaptation to manage the multimedia streams, in terms of efficiently allocating network resources to users based on their semantic preferences. In the ROI-based adaptation process, the high-level descriptions of video streams are adapted, steered by the automatically generated spatial metadata (see Section 6). The transcoding decisions are a trade-off between quality and compression, which can be different for ROIs and boundaries. In combination with information obtained from the QoS Manager, it is possible to reflect the (dynamically changing) bandwidth in the unequal partitioning of the data rate, devoting a higher rate to the ROIs while lowering the rate for the remainder of the video stream.

8. Conclusions

In this paper, a context-aware architecture for QoS of multimedia streams has been described. The main components are the context acquisition, the QoS and the multimedia adaptation subsystems. Context information is mainly devoted to user localization and network monitoring. The QoS Manager aggregates such information for the adaptation process. Adaptation is based on metadata descriptions of multimedia contents and consists of both transcoding and content manipulation.

Several tools are already available from different partners; current activities are aiming at their integration in a common experimental setup that will implement the full architecture described.

References

[1] Digital Living Network Alliance (DINA), URL: <http://www.dlna.org>.

- [2] Network of Excellence “Interactive Media with Personal Networked Devices”, FP6 EU Project, URL: <http://intermedia.miralab.unige.ch/>.
- [3] The UPnP Protocol Stack, URL: <http://www.upnp.org>.
- [4] K. Cheverst et al., “Developing a context-aware electronic tourist guide: some issues and experiences”, *SIGCHI Conference on Human Factors in Computing Systems*, 17-24, 2000.
- [5] A.K. Dey, “Context-aware computing: The Cyber Desk project”, *AAAI Spring Symposium on Intelligent Environments*, 1998.
- [6] P.J. Brown, “The stick-e document: A framework for creating context-aware applications”, In *Proceedings of Electronic Publishing*, Palo Alto, 259-272, 1996.
- [7] P. Baronti et al., “Wireless sensor networks: a survey on the state of the art and the 802.15.4 and Zigbee standards”, *Computer Communications*, 30:1655-1695, 2007.
- [8] H. Chen et al., “An ontology for context-aware pervasive computing environments”, *Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review*, 18(3):197-207, 2004.
- [9] R. Braden et al., “Integrated Services in the Internet Architecture: an Overview”, *RFC 1633*, June 1994.
- [10] S. Blake et al., “An Architecture for Differentiated Services”, *RFC 2475*, December 1998.
- [11] R. Braden et al., “Resource ReSerVation Protocol (RSVP)”, *RFC 2205*, September 1997.
- [12] B. Davie et al., “An Expedited Forwarding PHB (Per-Hop Behavior)”, *RFC 3246*, March 2002.
- [13] S. De Zutter et al., “System architecture for semantic annotation and adaptation in content sharing environments”, Accepted for publication in *The Visual Computer*, 2008.
- [14] H. Knoche et al., “The kindest cut: enhancing the user experience of mobile tv through adequate zooming”, *Proc. MULTIMEDIA 07*, New York, USA, 2007.
- [15] OSGi Alliance, URL: <http://www.osgi.org>.
- [16] The FP6 EU PERSONA project, URL: <http://www.aal-persona.org/index.html>.
- [17] R. Seacord et al., “Replaceable Components and the Service Provider Interface”, *Technical Note CMU/SEI-2002-TN-009*.
- [18] R. Morris et al., “The Click modular router”, *ACM Transactions on Computer Systems*, 18(3):263–297, Aug. 2000.
- [19] Application Layer Packet Classifier for Linux, URL: <http://l7-filter.sourceforge.net>.
- [20] D. Grossman, “New Terminology and Clarifications for DiffServ”, *RFC 3260*, April 2002.
- [21] ISO/IEC: *Information Technology – Multimedia Framework (MPEG-21) – Part 7: Digital Item Adaptation*, ISO/IEC JTC 1/SC 29/WG 11, December 2007.
- [22] M. Asbach et al, INTERMEDIA Multimedia Tools, URL: <http://sourceforge.net/projects/intermedia/>.