ISIS: INTELLIGENT SCALABILITY FOR INTEROPERABLE SERVICES

P.Gioia A.Cotarmanac'h	K.Kamyab P.Goulev E.Mamdani	I.Wolf A.Graffunder	G.Panis A.Hutter	A.Difino B.Negro	M.Kimiaei C.Concolato J.Dufourd	T.DiGiacomo C.Joslin N.Thalmann
France Telecom R&D, France	Imperial College London, UK	T-Systems Nova, Germany	Siemens, Germany	Telecom Italia Lab, Italy	ENST, France	University of Geneva, Switzerland

INTRODUCTION

With the growth of the fixed, wireless and mobile Internet, the convergence process of information technology, communication and media faces new challenges. The ISIS project was created to tackle some of these challenges. In particular, it proposes to solve one key issue: procure a technology to provide scalable, seamless multimedia services. Today's state of the art in the area of customized and personalized content for multi-devices and multi-networks delivery systems has validated the technical feasibility and the business perspectives of such technology, but it remains mainly limited to text-based data.

The goal of ISIS is to design, implement and validate a multimedia framework that allows for audio-visual content to be created once and adapted to a wide range of service scenarios or contexts by customization to different transport characteristics and end device capabilities as well as personalization to end-user preferences. The transport mechanisms envisaged are those of the fixed, wireless and mobile Internets. The end-devices include the variety of personal computers and handheld terminals connected to these networks. To this end, the following objectives were pursued: 1-Development of tools for coding, managing, delivering & playing back scalable audio, video & graphics contents; 2-Integration in a consistent end-to-end production, delivery and consumption chain; 3-Validation and demonstration of scalability functionality through a prototype application.

In this paper we present an overview of the work carried out on the ISIS project to date. We start by motivating our work. We then present an overview of the ISIS end-to-end multimedia delivery and consumption architecture. Within this section we give a brief description of the technologies developed in ISIS including the MPEG4 client-server architecture developed for both PC and IPAQ, personalization, adaptation including coding tools, and context awareness.

MOTIVATION AND BACKGROUND

The innovation of the ISIS project is a multimedia representation format and management tools that allow for audio-visual content to be created once and adapted to a wide range of service scenarios by customization to different transport characteristics, end-device capabilities, natural environment conditions as well as personalization to end-user preferences. These are referred to hereof as *usage environment*. This new technology will:

- enable content creators and providers to provide their content for many different *usage environments* without the need to create or store many different versions of the same content;
- enable service providers to provide their clients with the optimum quality and with customized content with respect to the users' current *usage environment*.

These objectives represent a significant progress compared to today's situation where:

- content and service providers have to develop, store and maintain multiple versions of the same content. It is then not always easy to choose the correct version to address all relevant user communities and relevant usage scenarios;
- end users have limited access to the content or very reduced QoS while accessing the content due to a mismatch between their available terminal and network resources and the content formats stored on the servers.

To achieve its objectives, the project has analyzed, specified and developed a complete end-to-end production chain which constitutes significant innovation over the state-of-the art in major technological areas of work:

1. development of innovative content representation formats with an emphasis on scalable content management techniques;
2. design of an integrated customization and personalization framework able to deal with complex multimedia content formats and metadata;
3. demonstration of the end-to-end platform accessing the content over various networks including fixed and wireless networks as well as fixed and mobile terminals.

© 2004 The Institution of Electrical Engineers.
Printed and published by the IEE, Michael Faraday House, Six Hills Way, Stevenage, Herts SG1 2AY, UK

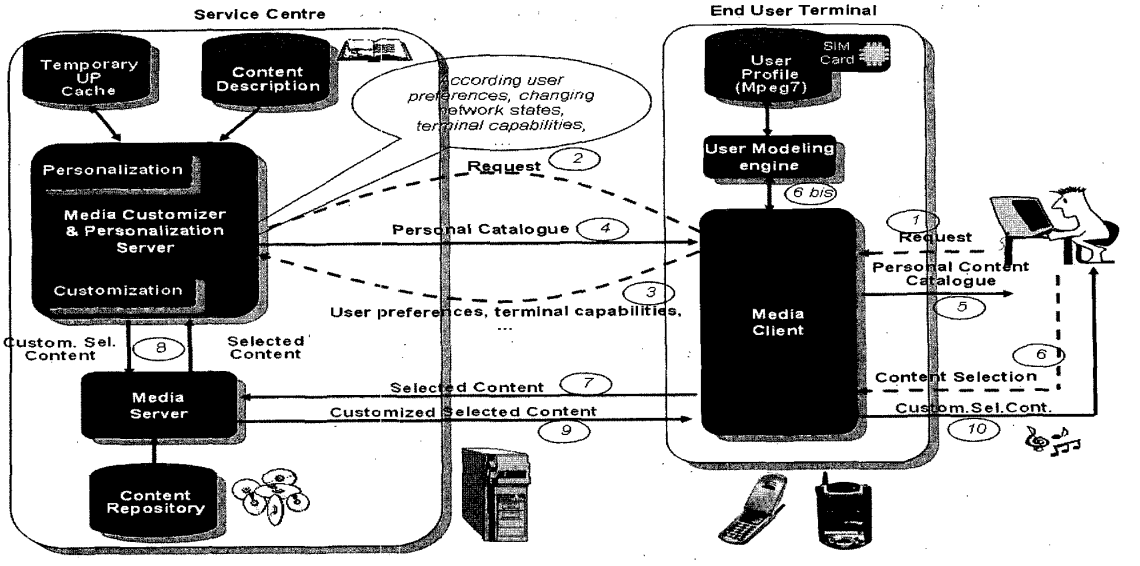

Figure 1: ISIS End to End Architecure

OVERVIEW OF THE ISIS FRAMEWORK

MPEG-4 Client-Server architecture for PC & IPAQ

One of the objectives of ISIS was to specify and develop a set of player tools to effectively play-back and interact with personalized and adaptive content among which still images, audio, video, vector graphics, animated virtual characters and scalable 3D meshes. Development occurred on both on a high-end terminal and on a handheld device such as the IPAQ. Given these requirements, namely the extensibility and the limited-resource platform targeted, the ISIS project has designed a modular architecture for content consumption tools based upon MPEG-4 and ISMA (Internet Streaming Media Alliance) specifications.

After evaluation of the complexity of codecs developed in the project as well as the available processing power and memory resources on handheld devices, the absence of native floating point operations, the memory footprint and the rendering capabilities were identified as being the main bottlenecks. As a consequence it was deemed appropriate to separate player tools according to their specific rendering requirements (2D A/V, 3D meshes or 3D animation renderers) which could then be optimized on each platform. As a result, the client side architecture consists of two layers : a control application layer and a player layer. The former is responsible for managing the interaction between a Personalizer module and several player tools (belonging to the player layer). The latter tools are :

- a 2D graphics audio and video MPEG-4 player,
- an MPEG-4 avatar player ,
- and finally a scalable mesh 3D player.

Once the architecture was defined and implemented, our main focus has been to improve the performance of the system. As an example, the implementation of 2D graphics on handheld devices has led to on-going major optimizations of the 2D rendering pipeline as well as to a thorough complexity analysis paralleling a new activity in MPEG on lightweight scene representations.

On the server side an ISMA compatible Real-Time Streaming Protocol (RTSP) based network stack has been implemented. A pure Java based design has been chosen in order to facilitate integration with the MPEG-21/DIA (Digital Item Adaptation) reference software which is completely written in Java. In order to achieve maximum extendibility, a plug-in interface has been designed which connects media specific service plug-ins with the streaming server kernel. Up to now four service plug-ins have been integrated in the server architecture:

- RawFileReader plug-in: Implements media streaming from file
- Customizer plug-in: Implements on the fly adaptation of streaming media according to the MPEG-21/DIA standard
- Transmoding plug-in: Implements on the fly transmoding of streamable graphics (cartoons) into MPEG-4 video
- Mesh3D plug-in: Implements an interactive 3D-terrain exploration service

Further work will include the implementation of an RTCP based bandwidth estimation tool into the existing client/server architecture allowing for on the fly

adaptation of streaming content according to network specific constraints.

Personalization

The field of information management has been undergoing a paradigm shift lately, from maintaining passive information warehouses which can be accessed by users all around the globe to one where the stress is on active customization of content to different users. This revolution can be best represented as the application of the concept of Personalized Information Services. In this domain, User Modeling is a technology that addresses the need to personalize information services on the Internet or in a similar network [1,12].

Personalization And User Modeling Approaches. There can be several ways to exploit the user model information in a personalized service, for example in the generation and fruition of a catalogue of multimedia content. Generally the user-modeling task can be done on either the server side or the client side.

The client side approach [2], as depicted in Figure 2, is known and excellent for a broadcast scenario, where all the contents and their descriptions are accessible to distribted users. This is the case of the digital and terrestrial TV, where TV events, through one or more aerial, reach variouos TV terminals (STB or other equivalent equipment and television screen). In this case, the module of personalization with the selection of a subset of interesting items and the module of management of the user model resides on the user's terminal. Client-side programs can obtain user information more easily and in a more timely fashion by observing user's behavior and learning from user's interaction with machine by means of a known device as a remote control. Moreover client-side user modeling can have a high degree of trust: the user has the complete control over his own user model as it is stored on his file system.

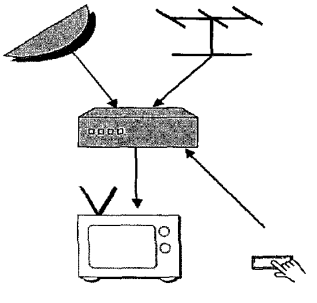

Figure 2: Client-Side User-Modeling Approach

Obviously, in a client-server scenario, where the information and description of personalized content reside on the server and are not broadcast over the network, the client-side user modeling approach is not appropriate or efficient [7]. But, not even the server-side personalization, as indicated in Figure 3, is perfect. In this scenario, all the relevant information about the user are kept and managed in an appropriate module in the server and the user has to register his profile in every new service and he has to trust that his personal data will be used effectively. In this case, the user's information is more difficult to obtain because the observed actions on the client terminal (telephone, hand-held computer, classic PC or other devices) have to be sent from the client to the server over the internet. On the other hand, placing the user model and the content database on the same machine may be helpful for indexing the database more efficiently based on the user model [3].

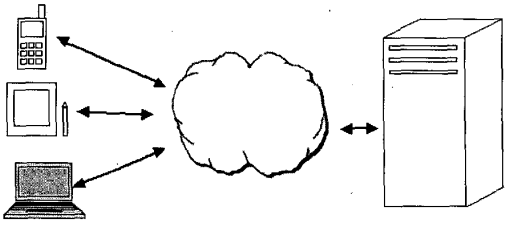

Figure 3: Server-Side User-Modeling Approach

In order to overcome the difficulties coming from the two previous approaches, the proposed solution contains a hybrid system of the above two paradigms. The following section will describe the new hybrid scenario.

Personalization and User Modeling in ISIS. From a static point of view, the complete scenario is composed of client equipment and by one or more servers. With reference to Figure 1, the client equipment can be a PC, a PDA, a hand-held computer, a mobile phone or other devices. On the client side, the proposed scenario makes provision for a Client Application.

In this proposed architecture, the User Model resides on the user's terminal. This User Model is stored and managed by the User Modeling Engine; this component can be more or less complex and accurate according to the characteristics of the user's equipment in terms of memory, CPU and so on. On the server side, the Personalizer Module is responsible for the generation of a personalized catalogue, filtering all the available items in order to propose only a significant subset to a user. This module on the server uses two distinct databases: the Users Database where the user models are stored in a temporary way and the Content Description Database that contains the descriptions of the available content.

Obviously, different servers and machines can host all these server components.

From a dynamic point of view, the user has at his disposal different types of equipment. We suppose, in this first interaction, that he's walking on the street and he wants to consult the lastest interesting and preferred content with his PDA. The client equipment contains a Client Application (for example the application that allows consultion of news) and an instance of the User Profile. This means that, on the client side, a User Modeling Engine manages and stores a complete User Model, according to the MPEG-21 standard. The User Modeling Engine works with all the known personal data (for example age, gender, sex, interests, hobbies, explicit preferences) and the usage history in the previous usage sessions. According to all these pieces of information, the User Modeling Engine is able, according to known techniques (if necessary merged together) to predict the best categories and characteristics for this user. For example, if the user is a successful man of 40 years of age and in his history he frequently watches financial and political news, the User Model Engine will predict that the user prefers News with categories Financial and Political and that he has a low interest in Gossip and Music News. When the Client Application contacts the Service Provider that provides this information service, the client sends a request (for example as an HTTP request, a SOAP message or other type of request) but it also sends, in a separate or aggregate thread, the user model or only one portion of it. In the simplest configuration, the client sends to the server only the preferences about the different managed categories (Financial News, Gossip, Music News, and so on), in a more complex scenario the user specifies his private information and the data that he agrees to diffuse.

When the server receives the user's information, it undertakes to cache the user model in an appropriate database only for the duration of this session. The user profile is volatile and, when the user will close the session, the copy of the user profile in the server will be deleted or, in alternative, it will be made anonymous. With this volatile version of user's information and user's preferences, the Personalizer Module on the server ranks all the available items in the database of Content Descriptions in a known way and generates a Personal Catalogue with all the best news for this specific user. The Personal Catalogue is sent from the server to the client. Finally the user views his catalogue on his terminal. Constantly monitoring the user's interaction, the client module can update, refine and improve the complete user model, with the aim of improving predictions at subsequent interactions.

Adaptation Framework.

Adaptation and MPEG-21. The ISIS adaptation framework realizes the fundamental MPEG-21 DIA [15] adaptation model, where an adaptation unit determines and applies optimal adaptation to the content based on the descriptions of the client's constraints and the content's scalability. For the application of the adaptation, the gBSD-based MPEG-21 DIA Resource Adaptation technology (shown below in Figure 4) is used.

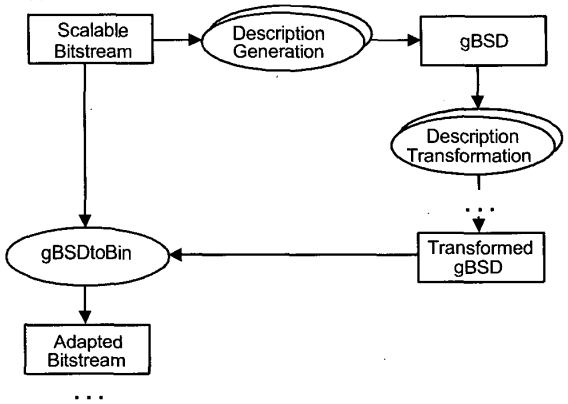

Figure 4: gBSD-based Resource Adaptation

The adaptation of a bitstream is abstracted through the use of an XML description of the bitstream, called generic Bitstream Syntax Description (gBSD). The gBSD can be transformed using XML transformation techniques, e.g. XSLT, to reflect the modifications that need to be applied to the bitstream in order to achieve adaptation. The Transformed gBSD is then processed to generate an adapted version of the original bitstream. This approach makes the adaptation process codec-agnostic and even allows multistage scenarios, where a bitstream is scaled down several times.

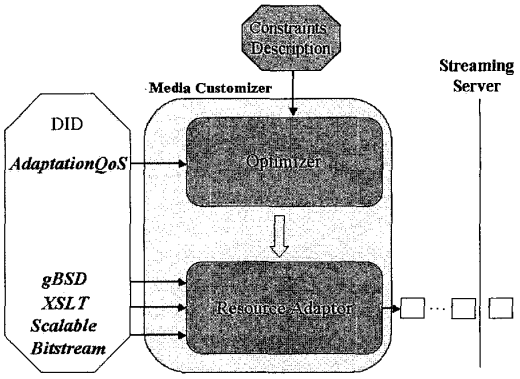

Figure 5: ISIS Adaptation Framework Overview

The ISIS adaptation framework, as shown in Figure 5, is centred on the Media Customizer, which is the framework's adaptation unit.

The Media Customizer consists of two logical modules, the Optimizer and the Resource Adaptor. The Optimizer retrieves and parses the Constraints Description and the content Digital Item Declaration (DID). The Constraints Description includes the terminal and network capabilities, described using the MPEG-21 DIA Usage Environment DS [15]. The DID carries all information about the available content and the necessary tools for adaptation, namely the AdaptationQoS, gBSD and XSLT. The Optimizer extracts the AdaptationQoS, a descriptor of possible adaptation options, which the Optimizer best tries to match to the given Constraints. The output of the analysis of AdaptationQoS and Constraints Description are specific values for the parameters needed by the XSLT. The Resource Adaptor retrieves the gBSD and XSLT from the DID and the parameter values from the Optimizer and applies the gBSD-based adaptation as shown in Figure 4. The generated data of the adapted bitstream is then packetised into RTP packets and passed to the Streaming Server for transmission to the client.

Scalable Audio. In the MPEG-4 Audio standard there exist several audio codecs for various purposes. In the Internet Audio Working Profile one can find the so called Advanced Audio Coding-Bit Sliced Arithmetic Coding (AAC-BSAC) which allows scaling of the encoded bitstream down to a certain base layer (typically 16 kbit/s) in 1kbit/s steps, see [16]. AAC-BSAC is the only fine granular scalable codec within MPEG-4 Audio.

BSAC encoding allows the use of several interleaving schemes depending on how many layers the bitstream is constructed of. To keep the example simplest we have chosen a layer structure which is basically one base-layer and 48 enhancement-layers representing the spectral information needed to decode one time frame. Base and enhancement layers plus some additional header information form an elementary stream that is presented below.

The bitrate adaptation in case of BSAC elementary streams is realized as truncation of the 48 enhancement layers. The base layers are not changed in terms of adjusting their length, but they include a parameter framelength, which needs to be updated during the bitstream transformation since it provides information necessary for the decoding process. The BSAC elementary stream starts with a BSAC header, which remains unchanged in the adaptation of the bitstream. The customization of such a bitstream can even be done in streaming scenarios frame by frame, since this unit is the minimum block a decoder needs to reconstruct the audio signal.

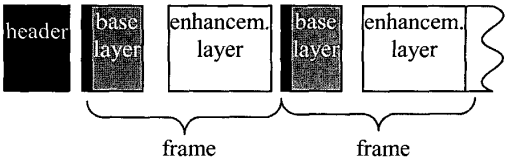

Figure 6: Basic Structure of the BSAC Elementary Stream Used in ISIS

BSAC solves the problem of exhaustive processing needed when transcoding of audio streams takes place nowadays. There is no need for decoding and again encoding with different parameters, since the scalability is based on the nature of the BSAC stream. In order to deliver an adapted stream, just a few blocks of the bitstream are truncated, while others are updated. This very processing efficient operation can be done on demand, just in the moment of the stream delivery as demonstrated within the ISIS project.

```
<gBSDUnit syntacticalLabel=":MA4:BSAC:BSAC_frame_element"
start="0" length="4168">
    <gBSDUnit
syntacticalLabel=":MA4:BSAC:BSAC_base_element" start="0"
length="392">
        <Parameter start="0" length="11"
name=":MA4:BSAC:BSAC_base_element:framelength" >
            <Value xsi:type="bt:b11">521</Value>
        </Parameter>
        <gBSDUnit start="11" length="5" />
        <Parameter start="16" length="6"
name=":MA4:BSAC:BSAC_base_element:toplayer">
            <Value xsi:type="bt:b6">48</Value>
        </Parameter>
        <gBSDUnit start="22" length="370"/>
    </gBSDUnit>
    <gBSDUnit syntacticalLabel=":BSAC_layer_element"
start="392" length="3776" marker="bitrate">
        <gBSDUnit start="392" length="24" marker="el1"/>
        <gBSDUnit start="416" length="24" marker="el2"/>
        <!--- and so on ...-->
        <gBSDUnit start="1456" length="24" marker="el47"/>
        <gBSDUnit start="1480" length="2688" marker="el48"/>
    </gBSDUnit>
</gBSDUnit>
```

Figure 7: Example gBSD Fragment for a BSAC Frame

Utilising the tools of the MPEG-21 framework. that is currently running the standardisation process, the BSAC audio can be automatically adapted according to the *usage environment.*

Scalable Video. A new family of wavelet based scalable video codecs is currently being discussed in MPEG. The central idea is to apply wavelet based filtering not only in the spatial domain but also in the temporal domain in order to achieve a high degree of energy compaction in all three dimensions, "Wavelet (2D+t)". It has been shown that the coding efficiency of the coding technique can be significantly improved when motion compensation is performed along the temporal dimension, [10].

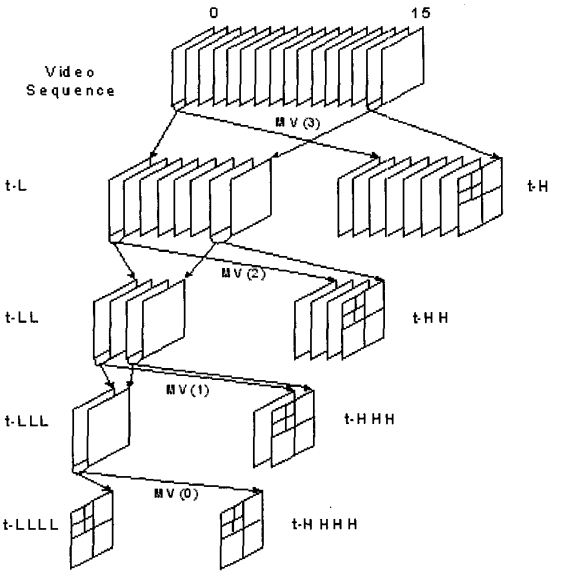

Figure 8: A Wavlet Based Four Level Decomposition

Based on the evidence from rate distortion experiments carried out in the ISIS project and the evaluation of other relevant criteria, the Wavelet 2D+t algorithm from [4] has been considered the most favorable solution for scalable video coding in ISIS.

The coding system divides consecutive frames into groups, similar to the Group of Pictures (GOP) structure in MPEG-1/2/4. Each GOP has a typical size of 16 frames. With reference to Fig. 6, the motion compensated (MC) filtering is performed on pairs of frames to produce (at level 1) temporal low- (t-L) and high-pass filtered (t-H) frames using the Haar wavelet. The temporal low frames are then decomposed again with the same MC filtering method. Continuing to level 4, the octave based five-band temporal decomposition shown in Fig. 6 is generated. In this case, we have 1 t-LLLL frame, 1 t-HHHH frame, 2 t-HHH frames, 4 t-HH frames, and 8 t-H frames per GOP (=16 frames here) and the associated motion vector fields MV(0), ... MV(3).

After the temporal filtering, a 2D spatial wavelet transform is applied to the temporally transformed frames. The resulting coefficients are then compressed by the EZBC spatial coder, and the fractional bit planes are interleaved for consistent quality within the GOP. The motion vector field is produced by hierarchical variable size block matching (HSVBM) and coded with a context based arithmetic code.

Due to the wavelet representation, which is used in the spatial as well as the temporal domain, the codec is inherently scalable along the spatial and temporal dimensions. However, the levels are restricted to powers of 2. In addition, fine granular SNR scalability is enabled by bit plane encoding. All scalability features can be mixed, e.g. it is possible to spatially and temporally scale a video sequence and at the same time fine scale the quality.

Furthermore, due to the embedded structure, the penalties in coding efficiency due to the scalability features are kept at a low level. Thus, the Wavelet 2D+t codec is a fully scalable and yet highly efficient video coding technique.

In the ISIS project a first optimization of the decoder code has been developed in order to support Wavelet 2D+t video coding in a complete end to end streaming chain. As a result, one of the first real-time decoder implementations at least for QCIF sequences at 25 fps has been developed. Currently, decoding of CIF images can be achieved at least at 12,5 fps on high-end PCs (2,8 GHz).

As a further result of the ISIS project tools have been generated which allow for the generation of gBSD data along with the encoding process and an XSLT style sheet has been developed which together enable MPEG-21/DIA compatible fine granular scaling of the video quality. Future work will extend these tools in order to support all 3 scaling modes, i.e. spatial, temporal and quality scaling.

Terrain Navigation (Wavelet Mesh 3D). The purpose of the 3DMesh scalable coding tool is to allow immersive virtual navigation in large scenes representing textured geographic data (see Figure 9) on various terminals. Due to the very large amount of data needed to represent such scenes, a scalable streaming format is needed. This format has to be associated with some transmission scenarios exploiting it fully. Indeed, the format described in this paper relies on geometric wavelets [9], coupled with a geometric adaptation of the classical Set Partitioning in Hierarchical Trees (SPIHT) zerotree-compression [11]. Though it provides very good compression and is naturally prone to adaptive decoding, the detail selection / navigation scenario has to be specified in a context of real-time client / server dialog in order to play "3D on demand" and navigate freely while the data is streamed.

In practice, the end user should be able to select a terrain database and, within a moderate time lap, flyover the terrain with no waiting time for transmission or loading. This is made possible by the selective transmission of the refinements needed by the current navigation [6]. While the observer is viewing a particular location, the client requests the needed data that will refine this location and setup the neighboring areas for future visualization. This concept is illustrated by Figure 10.

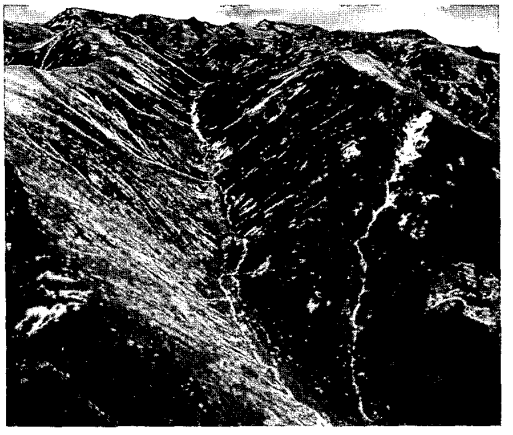

Figure 9: Textured Terrain During Flyover.

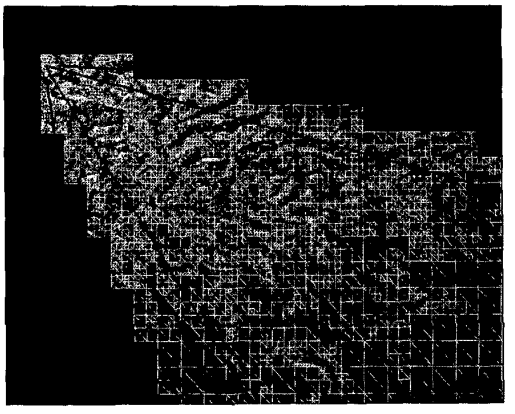

Figure 10: View From Above of a Virtual Navigation Over a Terrain. The Red Sector Represents the Actually Viewed Scene. The Blue Area is Composed of the Areas in Which New Data is Requested.

The overall behavior of both client and server is as follows: during navigation, the client maintains a cache with the information already transmitted. When a new location has to be refined, it sends its position and orientation in the scene to the server, along with compact information of its cache content. The server determines the local data and corresponding precision, and sends them to the client. This exchange is repeated periodically.

Besides the spatial adaptation, the underlying wavelet representation makes it possible to tune the level of detail at which the data are sent and displayed, indeed allowing the same scene data to fit various terminals and networks.

Adaptation of 2D Graphics. The kind of 2D graphics media targeted by the project are 2D graphics streams representing classic 2D cartoons, like many of the Flash cartoons that can be found on the web [13]. These cartoons are generally made of a static background and of characters moving on top of it. Each character is made of zones of uniform colors, sometimes filled with a gradient color or a bitmap texture. The different zones are surrounded with a border line of a different color and/or different width. In this kind of media, animation is the result of two processes: replacement or removal of elements of the previous frame by new graphical elements and motion of the camera or color transformations.

Within the scope of ISIS, a survey was made to analyze which technology best suited the needs of the project in terms of 2D graphics adaptation. It appeared that because of its binary format and associated features, and also because it is an open standard, MPEG-4 BIFS (BInary Format for Scenes) [14] was chosen to represent the graphics media of the project, using its standard XML representation format, called XMT.

Hence, we achieved an implementation of a BIFS encoder which enables the production of scalable 2D cartoons. The scalability techniques that we developed take advantage of the structure of a BIFS stream. A BIFS stream is made of a sequence of insertion, deletion or replacement commands. Therefore, a general scalable encoding technique consists in encoding the content into several logical layers of BIFS commands.

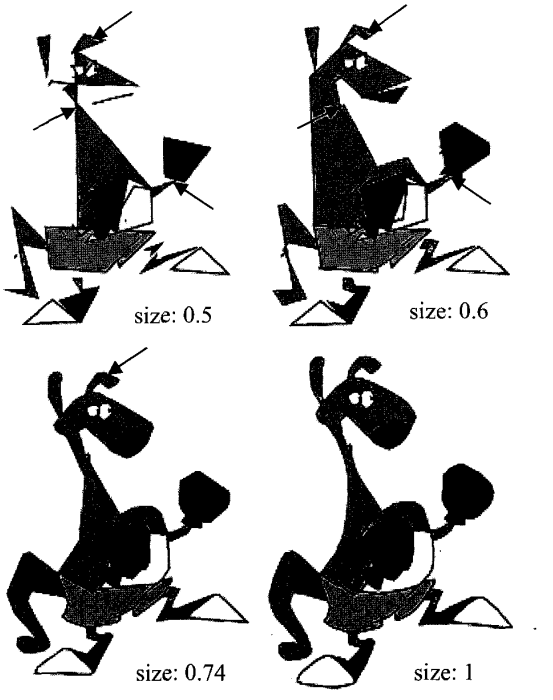

Figure 11: Quality Enhancement of 2D Graphics (top left to bottom right)

In line with the overall ISIS platform architecture, we designed a way to describe both the BIFS streams structures and the scalability hints using gBSD. The adaptation therefore consists in using these hints to drop some BIFS commands depending on the *usage environment* and on the parameter in the AdaptationQoS descriptor. An example of such an adaptation on a of the cartoon as is depicted in Figure 11. The top left image shows the base layer while the bottom right image shows the base layer with three enhancement layers. The quality enhancement are noticeable namely on the ears, the eyes, the wrists, etc.

Adaptation by Transmoding. To define transmoding, we first need to give an exact definition for "modality" and a list of modalities. In ISIS we have used the following hierarchical modality classification schema with five principal modalities and some sub-modalities:

- Video
- Audio (Audio2D, Audio3D, Speech)
- Image
- Graphics (Graphics2D, Graphics3D)
- Text

This Modality Classification Schema has reached a consensus in MPEG-21.

Transmoding is then defined as a change of modality based on the substitution of a Digital Item (DI) resource by an alternate versions of it in another modality.

The ISIS Transmoder plug-in consists of the following transmoding tools:

- Text to image.
- Graphics2D to video.
- Graphics2D to image.
- Video to graphics2D(slide show)

In order to realize adaptation by transmoding, we have added a Transmoding description tool to MPEG-21 DIA. Through this descriptor, any peer in the adaptation chain can express his preferences on the parameters of a specific conversion. A provider, for instance, may need

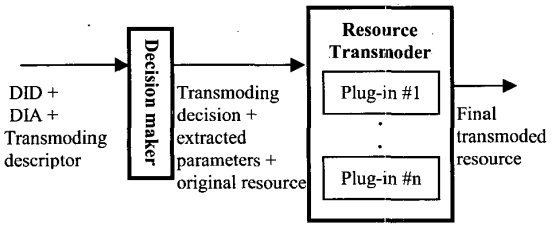

Figure 12: A Simple Architecture of Transmoder Module

to provide some transmoding hints to facilitate, guide or enable the adaptation, based on his knowledge of the resources available. The transmoding hints include the descriptions of the most general transformation parameters and are based on no particular underlying algorithm.

Figure 12 shows a very simple architecture for the ISIS Transmoder module. The Transmoder server plug-in is a module very similar to the Media Customizer and is then parallel to it in the overall architecture. The transmoder plug-in consists of two main blocks: Decision Maker and Resource Transmoder. The Decision Maker block, based on the given constraints decides on the optimal transmoding. The Resource Transmoder block performs the corresponding transmoding adaptation based on the transmoding parameters. The final output is the transmoded resource. The functionality of the Decision Maker block in the Transmoder plug-in is very similar to the functionality of the Optimizer in the Media Customizer, in the same way the function of Resource Transmoder block is equivalent to that of the Resource Adaptor.

The inputs of the Decision Maker block are all the constraints (DIA), original resources (DID) and transmoding descriptors. The output is the optimum transmoding decision. This decision, together with the extracted required transmoding parameters and the original resource to be adapted, constitute the input of the Resource Transmoder block.

Adaptation of 3D Virtual Humans. This section focuses on the adaptation process for 3D virtual humans. The main goal is to devise a method for multi-resolution representation/adaptation to transfer and render a virtual human (face and body) on heterogeneous target devices taking into account a wide range of complexity, compact data and animation for models, as well as interoperability and network capabilities. For more details, the reader can refer to [5].

We are using a clustered hierarchical model to define multi-resolution models that are scalable. The premise involves clustering all the data so that a specific complexity can be obtained by simply choosing a set of clusters. Each cluster has a set of vertices and vertex properties such as vertex normal, colors, and texture coordinates. Along with vertex information, the cluster has a set of indexed faces, normal faces, color faces and texture faces. Also each cluster can consists of sub-segments with its own material and texture. Each level is selected by choosing the appropriate blocks of clusters previously hierarchically constructed by half edge-collapsing operators. This selection is optimized by a pre-ordering of vertices data, taking advantage of the edge-collapsing. Compared to standard approaches [8], our proposed method is flexible and simple enough to

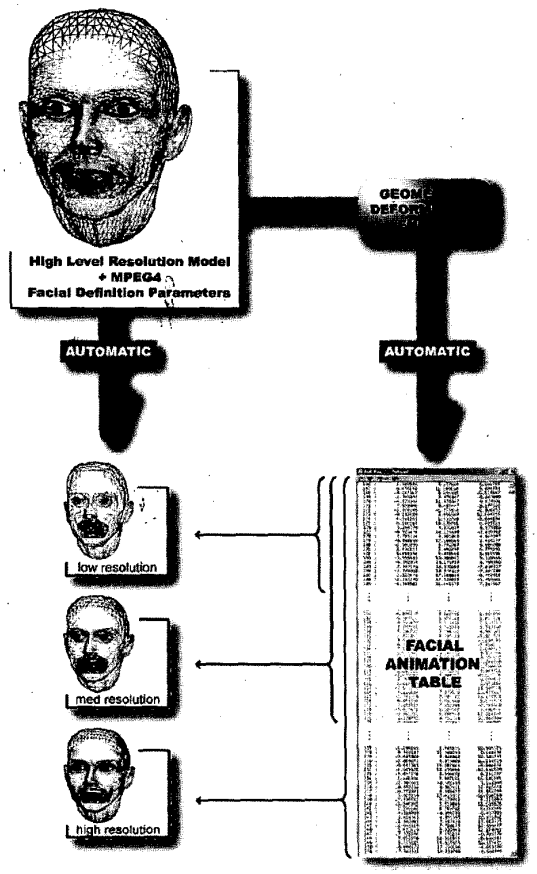

Figure 13: The Facial Animation Adaptation Process, Including Deformation Tables, Control Points and Face Representations.

allow efficient adaptation encodable to standard codecs. It also uses simpler adaptation mechanisms, similar to discrete level selection.

Animation data is applied to the adapted geometry resulting from the previous generation. For the facial animation, we use the MPEG-4 Facial Animation Parameters (FAP), defining each displacement with a piece-wise linear interpolation of FAP control points. From the high-level resolution model, we automatically construct the Facial Animation Table (FAT), which describes influences of FAP points and information for the interpolation. As the vertices of the model were ordered by influence, we can easily extract the corresponding FAT information from a high-level FAT table, for each model; therefore only the corresponding part of the global FAT table is then transmitted to the client. For facial animation, the global process of the adaptation preparation is illustrated by Figure 13. For body animation, we use MPEG-4 Bone-Based Animation (BBA) providing a generic and scalable framework. Correspondences between BBA nodes and H-ANIM joints are specified, as well as with pre-defined Level of Articulations for the adaptation of the BBA stream.

During adaptation, user's preferences generally consist of choosing explicitly a resolution, for the mesh and for animation, while terminal's capability is most complex, and is dependant on many factors. In this case we use device benchmarking, both for the digital item, tested at its highest resolution, and for the terminal, tested at initialization for instance. Finally, the adaptation due to network capacity is directly governed by the available bandwidth, the encoded file size, and the download time

Context Awareness and AffectiveWare.

MPEG-21 defines context as the *usage environment.* Although terminal, network and user characteristics can be known a priori, detected via software or learned via user modelling, natural environment characteristics need to be defined or detected on the fly. This is especially true for usage scenarios involving users accessing services on the go via mobile or handheld devices. Factors influencing natural environment characteristics are, for example, ambient light and noise. However, other factors can conceivably be included such as the user's social context (i.e. who they are with), mood, etc.

Contextual information about the natural environment can be gathered via sensors, both hardware based (physical) and user controlled (non-physical, i.e. GUI based). An agent-based context aware architecture has been developed that provides "plug and play" functionality for sensors providing context data to the ISIS architecture. Each sensor is monitored by a dedicated Sensor Agent that is launched by the Context Manager Agent (CMA) when the user selects a sensor. Sensor Agents are able to read data from both physical and non-physical sensors. The data is interpreted and provided to the CMA at specified intervals. In ISIS, the information gathered by the CMA is used in two ways. Firstly, sensor values can be queried at the time of requesting a multimedia presentation such as the multimedia catalogue and used to adapt the presentation. For example, readings from a light sensor can be used to adapt the colours and contrast used in the presentation. Secondly, it can be used to associate usage contexts with user actions. This information is then fed into the personalizer to influence future predictions. For example, if a user selects a specific content CN with context CX1, when a context CX2, similar to CX1, occurs then the personalizer may suggest CN to the user.

The AffectiveWare platform consists of hardware and software components which gather data from a series of physical sensors such as physiological sensors for skin conductance, but also light, temperature and so on. The AffectiveWare coder is able to communicate wirelessly with a PC or pocket PC via Bluetooth which makes it

ideal for on the go applications such as wearable computering and use with PDAs. Data produced by the AffectiveWare can be monitored by dedicated Sensor Agents which form part of the Context Management architecture.

CONCLUSIONS AND FUTURE WORK

The goal set by the ISIS project represents a considerable amount of work; the overall objectives for the project were to adapt several different media types towards unknown network, target device, and user constraints and personalization. The work presents a solid step from the concept of "multiple targets, multiple content" towards a paradigm of "multiple targets, single content". Within this framework, the following goals have been achieved:

- A modular client-server architecture, scalable to lightweight platforms.
- Adaptation modules based on MPEG-21 (for transmoding and general bitstream adaptation).
- Novel scalable codecs such as adaptive animated avatars and cartoons, scalable wavelet video coding, and wavelet 3D meshes as well as fine-grain scalable audio.
- A client-side user-modeling engine coupled with a state-of-the-art personalization server.
- Affectiveware platform coupled with a Context Manager providing pertinent contextual information and enhanced user experience.

Further work will be pursued in continuation projects such as DANAE with a strong focus on dynamic and distributed, in session, adaptation in a context-aware environment, with emphasis on scalable video coding.

ACKNOWLEDGEMENTS

The ISIS Project is a European co-funded project under IST-2001-37253 contract. We thank all members to the ISIS project whos' valuable contribution has made the work described possible. For more information on the ISIS project and its members please see [17].

REFERENCES

1. Baudisch, P., and Brueckner, L., 2002, "TV Scout: Lowering the entry barrier to personalized TV program recommendation."

2. Bellifemine, F. et al., 1999, Deliverable A12D1 "Agent-system software for the AVEB Phase 1 Demonstrator". FACTS (FIPA Agent Communication Technologies and Services) is the ACTS project number AC317 of the European Commission.

3. Buczak, A., Zimmerman, J. and Kurapati, K., 2002, "Personalization: Improving Ease-of-Use, Trust and Accuracy of a TV show Recommender". Proc. AH'02 Workshop on Personalization in Future TV, Malaga, Spain.

4. Choi, S., and Woods, J., 1999, "Motion-compensated 3-D subband coding of video," IEEE Trans. Image Processing, vol. 8, no. 2, pp. 155-167

5. Di Giacomo, T., Joslin, C., Garchery, S.. and Magnenat-Thalmann, N., 2003, "Adaptation of Facial and Body Animation for MPEG-based Architectures," IEEE CyberWorlds 2003.

6. Gioia, P., Aubault, O. and Bouville, C., 2002, "Real-Time Reconstruction of Wavelet-Encoded Meshes for View-Dependent Transmision and Visualization", ICIP 2002 Proceedings

7. Good, N., Schafer, J. B., Konstan, J. A., Botchers, A., Sarwar, B. M., Herlocker, J. L. and Ricdl, J., 1999, "Combining collaborative filtering with personal agents for better recommendations", Proceedings of the Sixteenth National Conference on Artificial Intelligence, pp. 439-446.

8. Hoppe, H., 1996, "Progressive Meshes," SIGGRAPH, pp. 99-108

9. Lounsbery, M., DeRose, T.D. and Warren, J., 1997, "Multiresolution Analysis for Surfaces of Arbitrary Topological Type", ACM TOG Vol. 16(1), pp. 34-73

10. Ohm, J., 1994, "Three-dimensional subband coding with motion compensation,", IEEE Trans. Image Processing, vol. 3, pp. 559–571

11. Said, A. and Pearlman, W. A., 1996, "A New Fast and Efficient Image Coded Based on Set Partitioning in Hierarchical Trees", IEEE TCSVT, vol 6, pp. 243-250

12. The Forrester Brief, TV Recommendation Engines, 2000, Beyond EPGs.

13. http://www.cartoonnetwork.com/watch/web_shows/

14. MPEG-4 Systems Specification, ISO/IEC 14496-1

15. MPEG-21 DIA Specification, ISO/IEC 21000-7

16. ISO/IEC FDIS 14496-3 subpart 4 Generic Coding of Audiovisual Objects

17. http://isis.rd.francetelecom.com/