A Framework for Utility-Based Multimedia Adaptation

Martin Prangl, *Student Member, IEEE*, Tibor Szkaliczki, *Member, IEEE*, and Hermann Hellwagner, *Member, IEEE*

***Abstract*—Content adaptation is an important issue of multimedia frameworks in order to achieve universal multimedia access (UMA), i.e., to enable consumption of multimedia content independently of the given resource limitations, terminal capabilities, and user preferences. The digital item adaptation (DIA) standard, one of the core specifications of the MPEG-21 framework, supports content adaptation considering a wide range of networks, devices, and user preferences. Most adaptive multimedia frameworks targeting the UMA vision do not consider utility aspects in their adaptation decisions. This paper focuses on a generic semantic-based audio–visual utility model for DIA that aims to enhance the multimedia experience for the user. Our proposed model is able to take the semantics and the perceptual features of the content as well as the users' specific utility aspects into account. Based on a detailed analysis of these constraints, we will show how the model reacts on individual input data. For choosing the best adaptation decision considering resource limitations on client and server sides as well as network characteristics, we evaluate four algorithms for performing this adaptation decision taking task. We will discuss results according to some use case scenarios.**

***Index Terms*—Adaptation decision taking, optimization problem, recommender system, semantic quality (SQ), universal multimedia access (UMA), universal multimedia experience (UME), utility model.**

I. Introduction

MULTIMEDIA services over computer networks are becoming widespread. The multimedia content can be delivered to different terminals such as desktop PCs, PDAs, and mobile phones. There has been a significant amount of research recently on the adaptation of multimedia contents to the actual usage context to ensure universal multimedia access (UMA) [1]. In many situations, the clients are unable to receive large audio–visual (A/V) data volumes in original quality because of resource limitations. Most multimedia frameworks try to comply with the capabilities and constraints of the user's terminal and do not consider the user him-/herself [2], [3]. However, the question "How to adapt multimedia data in order to provide the best user perceived utility?" is of central relevance and needs to be addressed.

To answer this question, physical issues such as terminal capabilities and network characteristics have to be considered. However, the quality of the adaptation significantly depends on the type and information content of the media as well. For example, it would be preferable w.r.t. universal multimedia experience (UME) [4] to adapt an action video in the spatial domain rather than in the temporal domain [5]. As a consequence, the user would get a smaller video window but he/she would still be able to fully enjoy rapid motion in action scenes. Therefore, the semantics of the content should be taken into consideration in the adaptation decision process. Moreover, especially in utility based adaptation frameworks, the semantic experience of a content should be optimized under given resource limitations. In this paper, we will introduce an adaptation decision model for DIA [6] which uses detailed *perceptual quality* (PQ) information and *semantic quality* (SQ) estimation. When considering quality in the multimedia area, we have to distinguish between its perceptual part and its semantic part [7]. The PQ is a metric about how a user perceives the content, and refers to the human visual system (HVS) [8]. For example, a smooth video has a higher PQ than a flickering one. The SQ, on the other hand, includes the designated information that the medium should convey to the user, e.g., the semantic content of a news report or the motion aspect of an action video [9].

Furthermore, there is a big difference between quality and utility in the area of multimedia applications. The term quality is mostly used to refer to the PQ whereas utility is a metric of overall satisfaction of the end user consuming this content. For example, if a user is consuming a video that is degraded because of resource limitations, the PQ, i.e., the visual impression, gets worse. The overall utility may be less degraded for the user if he/she is still able to extract the key information provided by the content. In this paper, the term utility refers to the overall user satisfaction, consisting of a combination of the perceptual and the semantic part of quality for the given content.

So called *cross-modal utility models* [10] are used to estimate the total utility of a media stream consisting of two or more modalities, e.g., video and audio. The total utility can be interpreted as a function which depends on the uni-modal utilities of the elementary streams themselves. In case of two modalities, namely video and audio, the total utility U can be defined as $U = f(U_V, U_A)$. U_V represents the video utility and U_A the utility of the audio stream. In the literature, there are some implementations of such a function; see, e.g., [11] for a discussion. All these implementations rely on adding the weighted uni-modal perceptual qualities, a multiplicative term (multiplication of uni-modal qualities), and specific constants in order to fit the subjective impressions of a group of test persons. The result of a detailed analysis of this approach [11] is that the implementation of the model itself as well as the weights and con-

Manuscript received July 14, 2006; revised October 17, 2006. This work was supported in part by the Hungarian State Eotvos Scholarship, in part by the Hungarian National Science Fund under Grants OTKA 42559 and 42706 and in part by the Mobile Innovation Center, Hungary.

M. Prangl and H. Hellwagner are with Klagenfurt University, 9020 Klagenfurt, Austria (e-mail: hermann.hellwagner@uni-klu.ac.at).

T. Szkaliczki is with the Computer and Automation Research Institute, Hungarian Academy of Sciences, 111 Budapest, Hungary.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2007.896650

stants are strongly dependent on the genre and the subjects participating in the test. For this reason, we see the lack of a more generic model for estimating the total A/V utility which can be used for any genre and which takes into account the individual user's preferences.

In our opinion, an approach for defining such a generic model has to start from the other direction. We avoid subjective perceptual testing to determine the model parameters for each content type, because this is expensive and time-consuming. Rather than giving a group of users a set of content variations for subjective testing, the individual user should be asked for his/her personal utility aspects. From these, such a generic model should be configured by fitting the model parameters to satisfy his/her individual preferences and utility concept. Of course, asking the user is critical in order to keep him/her unannoyed. For this reason, we applied a hybrid recommender approach, which is based on an easy-to-use feedback strategy and tries to configure the model based on the knowledge of previous user satisfaction. In case of the proposed utility model, high total utility should indicate high subjective PQ as well.

Another issue is to consider resource limitations on the server and client sides as well as the characteristics of the network links. Therefore, the resource requirements of the elementary media streams to be delivered (in our case, video and audio) have to be known before potentially adapting and combining them such that they fit the given resource limitations of the user's environment. The combination of (adapted) streams which complies with the resource constraints and which provides the best A/V utility value is the optimal solution for the consumer. Finding the "best" combination for the individual user within a reasonable (non-annoying) time frame can be seen as a challenging optimization problem.

The remainder of this paper is organized as follows. The next section gives an overview of our proposed multimedia framework enabling semantic-based A/V content adaptation. Then, the utility model used in our work is introduced in detail. Based on use cases, we show how it is possible to map high level user preferences and usage environment parameters to the SQ of a media stream. We introduce an automatic model configuration approach using a recommender strategy as well. Subsequently, the adaptation decision taking process is described as an optimization problem, considering resource constraints on the server, network, and client sides. We will present and discuss four different algorithms for solving this optimization problem. Finally, the runtime analysis of the adaptation decision task is presented and discussed.

II. Multimedia Framework With Audio–Visual Utility Modelling

Fig. 1 shows the concept of the proposed approach and its integration into a multimedia framework. The given user preferences and the genre of the requested content (influencing SQ) have to be known for configuring our generic utility model [12] which is used by the adaptation decision taking engine (ADTE) [13]. Currently we are distinguishing between five main genre categories in our experimental system: *action*, *news*, *cartoon*, *docu* and *sports*. This input information is mapped to specific model parameters which we call *high level adaptation parameters*, discussed in Section III. The individually configured model additionally needs to know the PQ of all deliverable content variations. The video variations can be characterized by the values of their spatial resolutions, frame rates and SNR variations. The audio variations may be distinguished by the values of their bit rates, sample rates and number of audio channels. Based on the genre and PQ information, the total utility U of all deliverable A/V variations can be estimated. Having available the utility and the information about the media-specific required resources (e.g., needed bit rate) of each deliverable A/V variation, as well as the information about the available resources on the client and server sides (e.g., the available bandwidth, battery status, or CPU power), the ADTE is then able to estimate the optimum adaptation strategy for the given content request [13]. This task has to be done quickly in order to avoid annoying media startup delays. This is required such that the ADTE becomes able to request individually the calculated utility value of a specific A/V variation under consideration. A detailed discussion on this issue is given in Section V. The found optimal adaptation decision is expressed by a set of parameters which we call *low level adaptation parameters*. They define an A/V media stream variation by its features (e.g., frame rate, spatial resolution, sample rate). Based on these target features, the adaptation engine (AE) performs the adaptation step on the original content. Finally, the produced variation, fitting the user's preferences and the usage environment and providing the best possible utility under the given conditions, can be delivered for consumption to the requesting client. Note that it is not possible that the ADTE selects a variation that the AE cannot produce because the ADTE has information about the AE capabilities as well.

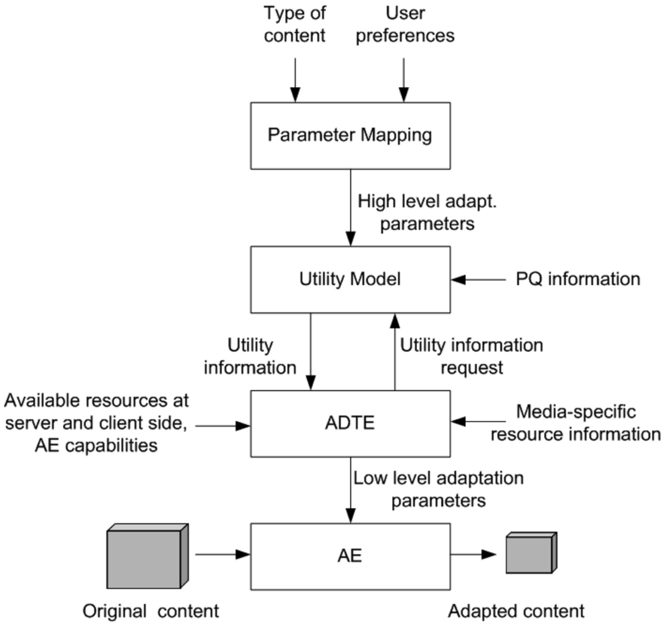

Type of content
User preferences
Parameter Mapping
High level adapt. parameters
Utility Model
PQ information
Utility information
Utility information request
Available resources at server and client side, AE capabilities
ADTE
Media-specific resource information
Low level adaptation parameters
AE
Original content
Adapted content

Fig. 1. Overview of integrated multimedia framework with A/V utility modelling.

III. Utility Model

The basis of the proposed model is that the total utility U_E of an elementary stream E can be split up into a perceptual part and a semantic part [7] as follows:

$$U_E = s \times \mathrm{PQ} + (1 - s) \times \mathrm{SQ}$$

where $s \in [0, \ldots, 1]$ denotes a weight that indicates how much influence PQ has on the total utility. Because SQ is the most important part indicating how the user receives the designated content information, we considered s in a range between 0.1 to 0.5. We compared several results by adjusting this weight and found an appropriate influence of the SQ specific, high level parameters by defining $s = 0.3$ in our model implementation. Note that PQ and SQ are normalized, i.e., in the range between 0 (worst) and 1 (best value). In the A/V case, we have to merge the utilities of the video and audio parts as follows:

$$\begin{aligned} U = \alpha \times [s \times \mathrm{PQ}_A + (1 - s) \times \mathrm{SQ}_A] \\ +(1 - \alpha) \times [s \times \mathrm{PQ}_V + (1 - s) \times \mathrm{SQ}_V]\,. \quad (1) \end{aligned}$$

PQ_V and PQ_A are representing the PQ of the video and the audio part, respectively, and SQ_V and SQ_A represent the corresponding semantic qualities. A multiplicative term and an additive constant, as used in perceptual cross-modal quality modelling [11], is omitted in our A/V utility approach. The reason is that we do not use a regression analysis based on subjective tests. Furthermore, the multiplicative cross-modal perceptual influence would be negligible in our case because our model is strongly bound on additive uni-modal semantic aspects. α denotes the importance weight of the audio modality. For example, in the case of a newscast, the importance of the audio part would be higher than the video part, resulting in a high value of α. It represents a high level adaptation parameter. All high level parameters act as weights and are directly depending on the content type (genre) and individual user preferences. Note that we do not (yet) use detailed low-level content feature extraction and analysis as introduced in [14]. Our current goal is to create the basic model and the framework and assess their usefulness and "performance." A path to improve the system in the future is certainly to take more detailed scene features like motion and object information into consideration within the parameter mapping process. This would enable a better, more fine-grained classification of the content than using the type of content alone for this task. The genre and other content specific information can be provided by MPEG-7 descriptions [15]. The user preferences as well as the usage environment can be easily described by MPEG-21 DIA usage environment descriptors (UED) [6] for interoperable exchange.

For perceptual video quality estimation we use the *"General"* model according to ANSI T1.801.03-2003, provided by VQM-Software,[1] and for audio PQ estimation we use the well known PEAQ metrics [16].

It is not suitable to do PQ estimation online due to the high computational requirements. However, the offline PQ results can be provided by the MPEG-21 DIA adaptation QoS (AQoS) descriptor [17]. The correlation between objective PQ metrics and the HVS is still low [18]. This fact implies the question: What is the "best" adapted variation for the individual end user? The answer is that this depends on the semantics of the content, the information which the user should receive by consuming the media stream. This semantics can be derived from the genre and the corresponding importance of the low level adaptation parameters. For example, in case of an action video delivered under bandwidth limitations, the semantic experience would be higher if the video were adapted in the spatial domain than in the temporal domain; i.e., the spatial resolution should be reduced and the frame rate of the original video should be kept intact. This adaptation step would result in a smaller window, but retain high motion in the video.

This consideration leads us to the definition of the relative value of the semantic content of an individual elementary media stream $\mathrm{SQ}_E = f(W, F)$ where W is a set of individual high level parameters (user and genre specific) and F represents a set containing data indicating the degradation of each feature in the stream (content variation specific).

The definition of the semantic video quality SQ_V is given in (2). The high level parameters w_{Fv}, w_{Sv}, and w_{Qv} act as importance weights of the video stream features. Note that the unique stream features of the video variation are nothing else than the low level video adaptation parameters. fr, $height$, $width$, and q represents the frame rate, the spatial resolution, and the quantization parameter of the video variation, respectively. $q_{\min}$ and $q_{\max}$ represent the codec (or AE) specific minimum and maximum quantization values. fr_{orig}, $height_{\mathrm{orig}}$, and $width_{\mathrm{orig}}$ are constants representing the corresponding features of the original video stream.

The resulting SQ_V points form a plane in the stream feature space, where the high level parameters act as weights defining the slope of the plane

$$\begin{aligned} \mathrm{SQ}_V = {} & w_{\mathrm{Fv}} \frac{fr}{fr_{\mathrm{orig}}} + w_{\mathrm{Sv}} \frac{height}{height_{\mathrm{orig}}} \frac{width}{width_{\mathrm{orig}}} \\ & + w_{\mathrm{Qv}} \left(1 - \frac{q - q_{\min}}{q_{\max} - q_{\min}}\right) \\ & w_{\mathrm{Fv}},\ w_{\mathrm{Sv}},\ w_{\mathrm{Qv}} \epsilon [0, \ldots, 1],\ w_{\mathrm{Fv}} + w_{\mathrm{Sv}} + w_{\mathrm{Qv}} = 1 \\ & fr \leq fr_{\mathrm{orig}} \\ & height \leq height_{\mathrm{orig}} \\ & width \leq width_{\mathrm{orig}} \\ & q \epsilon [q_{\min}, \ldots, q_{\max}]. \end{aligned} \quad (2)$$

The definition of the semantic audio quality again relies on a weighted approach of its modality features like sample rate (sr), encoding bit rate (abr) and the number of channels $(achan)$, which is given in (3). The high level parameters w_{Sa}, w_{Ba}, and w_{Ca} act as importance weights of the audio stream features.

$$\begin{aligned} \mathrm{SQ}_A = {} & w_{Sa} \frac{sr}{sr_{\mathrm{orig}}} + w_{Ba} \frac{abr}{abr_{\mathrm{orig}}} + w_{Ca} \frac{achan}{achan_{\mathrm{orig}}} \\ & w_{Sa},\ w_{Ba},\ w_{Ca} \epsilon [0, \ldots, 1],\ w_{Sa} + w_{Ba} + w_{Ca} = 1 \\ & sr \leq sr_{\mathrm{orig}},\ abr \leq abr_{\mathrm{orig}},\ achan \leq achan_{\mathrm{orig}} \end{aligned} \quad (3)$$

Increasing the weight of a parameter increases its importance. Yet, it does not follow from this that the same weight for two parameters results in their equal importance because some parameters may lead to higher utility within the same resource limit. For example, an audio stream needs much less resources than a video stream. For this reason, if we choose $\alpha = 0.5$ then we can reach higher utility preferring the audio stream under resource limitations.

In order to show the reaction of the proposed A/V utility model, we captured four typical A/V content scenes from high quality digital television (DVB). The first one was an action scene of *Stargate*, the second a soccer game *(Rapid Vienna versus Juventus Turin)*, the third a talking head news clip *(n-tv)* and the fourth a nature documentation about an octopus (*Universum*). All of them are original MPEG-2 A/V

[1] http://www.its.bldrdoc.gov/n3/video/vqmsoftware.htm

TABLE I
INDIVIDUAL MODEL CONFIGURATIONS AND CORRESPONDING ADAPTATION DECISION RESULTS FOR AN AUDIO–VISUAL ACTION SCENE FOR TWO DIFFERENT BANDWIDTH CONSTRAINTS

Resource	High level model configuration							Resulting low level adaptation parameters						Quality and utility			
bw limit	α	w_{Fv}	w_{Sv}	w_{Qv}	w_{Sa}	w_{Ba}	w_{Ca}	$w \times h$	fr	q	sr	abr	$achan$	PQ_V	PQ_A	U	MOS
250	0.1	0.2	0.4	0.4	0.2	0.8	0.0	720x576	1	23	48000	64	1	0.996	0.341	0.646	3
250	0.1	0.8	0.2	0.0	0	0.2	0.8	176x144	25	10	44100	96	2	0.816	0.644	0.807	6.483
250	0.5	0.2	0.2	0.6	0.2	0.8	0.0	176x144	25	10	44100	96	1	0.816	0.644	0.568	6.414
250	0.8	0.2	0.2	0.6	0.4	0.6	0.0	176x144	18	10	48000	128	1	0.619	0.708	0.537	6.31
250	0.9	0.2	0.8	0.0	0.2	0.8	0.0	720x576	1	26	48000	128	1	0.994	0.708	0.404	3.034
250	0.9	0.2	0.8	0.0	0.4	0.4	0.2	720x576	1	26	48000	128	2	0.994	0.708	0.656	3.276
250	0.95	0.4	0.4	0.2	0.0	0.8	0.2	176x144	5	10	44100	192	2	0.452	0.829	0.387	4.966
350	0.1	0.2	0.4	0.4	0.2	0.8	0.0	720x576	1	19	48000	64	1	0.997	0.341	0.695	2.414
350	0.1	0.8	0.2	0.0	0	0.2	0.8	320x200	25	29	44100	96	2	0.825	0.644	0.822	7.103
350	0.5	0.2	0.2	0.6	0.2	0.8	0.0	176x144	25	10	44100	192	1	0.816	0.829	0.596	6.172
350	0.8	0.2	0.2	0.6	0.4	0.6	0.0	176x144	25	10	48000	192	1	0.816	0.778	0.574	6.552
350	0.9	0.2	0.8	0.0	0.2	0.8	0.0	720x576	1	24	44100	192	1	0.995	0.829	0.426	2.793
350	0.9	0.2	0.8	0.0	0.4	0.4	0.2	720x576	1	24	48000	192	2	0.995	0.778	0.675	2.724
350	0.95	0.4	0.4	0.2	0.0	0.8	0.2	176x144	25	10	44100	192	2	0.816	0.829	0.404	6.138

TABLE II
INDIVIDUAL MODEL CONFIGURATIONS AND CORRESPONDING ADAPTATION DECISION RESULTS FOR AN AUDIO–VISUAL SOCCER SCENE, WHERE THE MAXIMUM LINK BANDWIDTH $\mathrm{bw} = 250$ kb/s AND THE SCREEN SIZE OF THE TERMINAL IS LIMITED TO 640 × 480 PIXELS

High level model configuration							Resulting low level adaptation parameters						Quality and utility			
α	w_{Fv}	w_{Sv}	w_{Qv}	w_{Sa}	w_{Ba}	w_{Ca}	$w \times h$	fr	q	sr	abr	$achan$	PQ_V	PQ_A	U	MOS
0.7	0.6	0.4	0.0	0.4	0.4	0.2	176x144	24	31	48000	96	2	0.228	0.552	0.629	5.793
0.1	0.6	0.2	0.2	0.2	0.8	0.0	176x144	25	30	48000	64	1	0.426	0.231	0.663	5.517
0.5	0.2	0.4	0.4	0.2	0.2	0.6	176x144	3	17	48000	96	2	0.382	0.552	0.612	3.586
0.6	0.0	1	0.0	0.6	0.2	0.2	640x480	1	31	48000	128	2	0.137	0.713	0.762	2.551
0.1	0.2	0.6	0.2	0.2	0.8	0.0	640x480	1	27	48000	64	1	0.146	0.231	0.545	2.517
0.1	0.2	0.2	0.6	0.2	0.8	0.0	176x144	2	10	48000	64	1	0.348	0.231	0.604	2.483
1	0.0	0.2	0.8	0.0	0.8	0.2	320x200	1	24	44100	192	2	0.053	0.856	0.429	2.172

streams and need a bit rate of 5–6 Mbit/s with the following features: $width_{\mathrm{orig}} = 720$, $height_{\mathrm{orig}} = 576$, $fr_{\mathrm{orig}} = 25$ fps, $sr_{\mathrm{orig}} = 48$ kHz, $abr_{\mathrm{orig}} = 192$ kb/s, $achan_{\mathrm{orig}} = 2$. In order to be able to compare the model behavior with results from subjective tests, the duration of each scene was limited to 10 s. We applied seven different model configurations to each scene in order to evaluate the model behavior under diverse "use case scenarios," which are basically different usage environment constraints. The evaluation for the action video is given in Table I for two different bandwidth limitation constraints. An additional use case constraint was that the AE has only one video and one audio codec (H.264 and MP3) available and is only able to handle 21 quantization levels from $q_{\min} = 10$ to $q_{\max} = 31$. Column 1 represents the bandwidth limit (kb/s), columns 2–8 the high level parameters (model configuration), columns 9–14 the resulting low level parameters. On the right-hand side of the table, the PQ values for the audio and video part as well as the total calculated utility U, and the subjective mean opinion score (MOS) values are given. Note that the utility value U is used within the adaptation decision taking process to select the variation providing the highest value for the specific high level parameters.

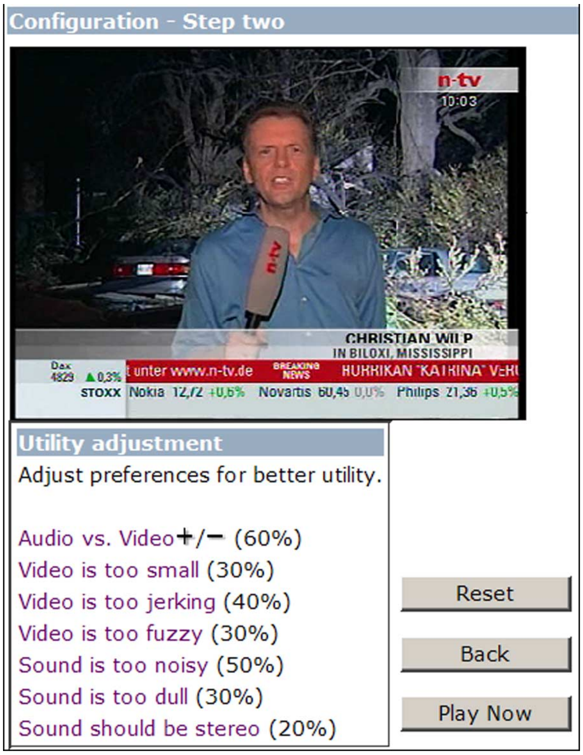
We applied the adaptation decision taking algorithms presented in Section VI to determine the low level parameters. The subjective MOS values were obtained by absolute category rating (ACR) according to ITU.T Rec. P.910 on an 11–grade scale (from bad to excellent). Thirty test persons (students from Klagenfurt University, nonexperts) were invited to rate seven degraded A/V output variations for the four content types at three different bandwidth limitation constraints (150, 250, and 350 kb/s), 84 variations in total. The duration of one subjective test program was 25 min on average and did not exceed the critical limit of 30 min. The tests were performed in the Usability Lab of Klagenfurt University, all test person had the same external environment in terms of brightness and noisiness. The visual modality was displayed on a TFT monitor at a distance of 50 cm in front of the user, the audio was provided by headphones.

It is seen from the table that the second configuration is the most appropriate for action video among the examined ones because it has the highest MOS in the corresponding lines. We selected a relatively large weight for the frame rate there. In accordance with this selection, variations with the high frame rate have high utilities at both bandwidth limits.

Another use case example is given in Table II. A soccer video should be delivered at 250 kb/s to a client device with a screen size of 640 × 480 pixels. The AE at the server side has the same quantization restrictions as in the previous example. For the soccer scene, the first configuration proved to be the best, i.e., where the weight for the frame rate w_{Fv} is large and the audio is also important (α is high).

IV. RECOMMENDER-BASED MODEL CONFIGURATION

The high level parameters for the use cases introduced in the previous section were applied based on intuitive, hand-crafted rules saying, e.g., if we have an action video, we apply a relatively high weight for the frame rate. These well defined rules should increase the semantic experience for the consumer. But the question is: Are these rules valid for an individual user? Our performed subjective MOS estimation as well as related subjective PQ tests [19] show that users have different tastes. In our case, applying the same parameters for each user would lead to different multimedia experiences for the individuals. However, the aim of the utility-based multimedia framework is to offer a personalized version of the content that leads to the optimum utility for the individual requesting the content. As al-



Adjust preferences for better utility.

Audio vs. Video+/− (60%)
Video is too small (30%)
Video is too jerking (40%)
Video is too fuzzy (30%)
Sound is too noisy (50%)
Sound is too dull (30%)
Sound should be stereo (20%)

Fig. 2. Prototype preview critiquing user interface.

ready mentioned, the framework takes the individual user preferences into consideration as well. Asking the users for their preferences, i.e., asking them to indicate the high level parameters of the model, would be easy. But in general, the users are not experts in the multimedia domain and do not know the optimal settings for the requested content and the actual environment in advance. Furthermore, the user would get annoyed if he/she had to answer too many questions in order to provide this very helpful information for the system. For this reason, it would be helpful for the user, if he/she would get a recommendation from users which consumed the same type of content under similar environment conditions. So called *recommender systems* [20] try to offer the individual user only those items which aim to be useful or interesting for him/her. Such systems are well known in other domains, e.g., information retrieval systems, online shops, and financial or insurance services. So called *collaborative filtering* (CF) recommender strategies predict the interests of a user by collecting taste information from many users [21] based on well-defined feedback mechanisms. The underlying assumption of the CF approach is that people who agreed in the past tend to agree again in the future. This strategy forms the basis of our implemented recommender approach. Pure CF systems have the drawback that so-called ramp-up problems can occur if there is not enough user feedback for individual items [22]. In this case, a sensible recommendation is not possible. For this reason, *hybrid strategies* are developed for specific problem domains [23]. Their aim is to combine the advantages and avoid disadvantages of specific strategies as far as possible.

Our approach for such a hybrid recommender system suggesting the individual high level parameters is a combination of a CF and a so called *knowledge-based* recommendation strategy. Knowledge-based recommender systems [24] rely on a domain-specific knowledge base containing rules. These rules indicate which item is suitable for which user request. The results are usually calculated by solving so-called *constraint satisfaction problems*, which are well known in the AI research area [25]. In our system such rules are, for example in case of a music video, that the priority of the audio modality is high, or for a newscast, that the number of audio channels becomes less important. More detailed audio and video specific rules rely on related experiments [5].

In order to find suitable similarities of users, we use individual demographic features and personal interests. For demographic feature information, we use a user's age and the gender, since older and young people may have different utility aspects, and males and females usually have distinct preferences as well. Furthermore, the individual favorite genres are used to estimate user similarities because users are more attentive watching interesting content in general. In our system, all information about a new user is captured by a user registration process. However, the user is able to change his/her preferences (e.g., favorite genres) any time.

The most important information of a recommender strategy in general is *feedback*. Feedback is needed in our case to learn about the users' utility "tastes." The chances to give the user an accurate utility model configuration grow with the number of rated utility impressions. Feedback is also important for finding similarities between users. There is a high probability that users who gave many similar ratings in the past, will also share utility impressions in the future. In order to keep the degree of disturbance for the user low, we use a special type of feedback strategy, where the user *critiques* the received utility of the consumed media presentation. For that purpose, the user is asked for his/her individual utility satisfaction after consumption of a five second preview session by choosing so called *tweaks* [26]. Tweaks are well defined statements the user can choose, e.g., *"The video was too small," "Sound was too noisy,"* etc. Fig. 2 shows the prototype preview critiquing interface of our implemented system. The chosen tweak causes an adjustment of the corresponding high level parameter, e.g., the tweak *"The video was too jerking"* causes an increase of w_{Fv} and a decrease of w_{Sv} and w_{Qv}. The user is able to get an updated preview variation or if the user is satisfied he/she is able to consume the whole content variation. Note that the user is not obliged to critique the content preview. He/she is able to consume the suggested content variation without preview feedback as well. However, at the end of the whole content consumption, the user is asked to rate his/her overall utility satisfaction on an 11–grade MOS scale. In order to assess and improve the convenience and ease of use of the feedback mechanism for the user, it would be useful to evaluate the system by means of usability tests. However, again, our focus currently is on the basic framework and optimizations like this are the scope of future work.

The user ratings as well as the proposed user- and content-specific data are stored in a database. The user similarity for the CF part is calculated based on the user's environment (dark, bright, loud, etc.), the network connection (Cable, DSL, LAN, WLAN), the terminal type (PDA, mobile, etc.), the user's demographic features and interests in content types as well as the user's rating history using the Pearson correlation [27]. The model configuration which got the highest rating from all the user's neighbors is selected for the actual request. If a user is requesting a specific content which was not rated by a "near" neighbor or if the

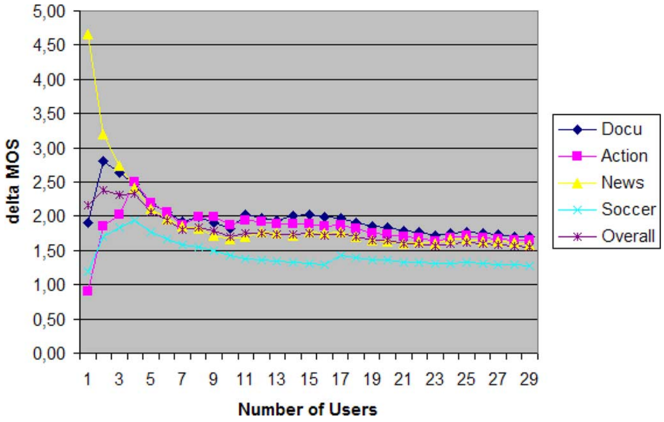


Fig. 3. Average difference of predicted MOS and user rated MOS in dependence of the number of involved users.

nearest neighbor is "too far away," the high level parameters are estimated by using the predefined rule base only.

In order to evaluate the success of the recommender approach, we took the MOS results earned from subjective tests for all four content types. We "replayed" the 30 users (test candidates) doing their MOS ratings in the same order as in the actual subjective tests. For each user, the recommender system predicted the model configuration based on related data and the corresponding MOS value. Fig. 3 shows the mean absolute difference (delta MOS) of the predicted MOS values and the real user rated MOS values for all tested content types. The mean overall error of all content types is included as well. The average MOS error is decreasing with the number of rating users for all four content types. The decrease is not monotonic but it indicates that the predicted model configurations get more reliable with the number of known ratings.

V. Problem Model of Adaptation Decision Taking

Based on the proposed utility model, which is individually configured for the user, the ADTE has to choose the most appropriate adaptation parameters w.r.t. the actual resource limitations, terminal capabilities, user preferences as well as AE capabilities (e.g., codec types, quantization levels, etc.) in order to provide the maximum media utility to the user. This process has to be performed very fast in order to keep the startup delay of the requested session in a non-annoying range. Especially under dynamic resource limitations, e.g., network bandwidth fluctuations, this decision has to be taken in real-time in order to provide continuous delivery to the client. This section develops the optimization problem model derived from the above utility model.

A client is requesting a movie m from the streaming media server. The original movie consist of one video and one audio stream. Both the video and audio streams can be adapted into uniquely defined variations, characterized by a set of video features F_v and a set of audio features F_a. They together form the feature set of a movie, denoted by F_m, which can describe the variations of the movie: $F_m = F_v \cup F_a$. The features can be, e.g., spatial resolution, frame rate, type of codec, number of audio channels, and audio sampling rate. Let features $f_1, f_2, \ldots, f_n$ denote the features ($n = |F_m|$).

Let V_v, V_a denote the sets of deliverable video and audio variations of movie m on the server (w.r.t $\sim$ the AE capabilities), respectively. Let V_m denote the set of deliverable variations of movie m. Let M and N denote the number of the different video and audio variations, respectively: $M = |V_v|$, $N = |V_a|$. The video and audio streams can be combined arbitrarily into a movie, that is, $V_m = V_v \times V_a$. $m|_f$ denotes the value of the feature f of stream m. The particular movie, video and audio variations are denoted by v_m, v_v and v_a, respectively. The variations can be specified as vectors in the feature space: $v = (k_1, k_2, \ldots, k_n)$. A client request on the movie consists of acceptance sets A_f for each feature $f \in F_m$ which can be acceptance ranges $[f_{\min}, f_{\max}]$ as well as a special case. Values of the features of the delivered stream m have to fall into the given acceptance sets. If the values can be sorted in A_f then let $g_{f,i}$ denote the ith smallest value in A_f. $g_{f,i_1} > g_{f,i_2}$ if $i_1 > i_2$. $l(f_k)$ is the number of different available and acceptable values of feature f_k.

Furthermore, the utilities of each deliverable video and audio variation are known or can be calculated. Let $U_V(v_v)$ and $U_A(v_a)$ denote the utilities for video variation $v_v \in V_v$ and audio variation $v_a \in V_a$, respectively. As already mentioned, the utility of the multimedia stream resulting from the combination of the video and the audio streams can be calculated as a weighted sum of the utilities of the two modalities: $U(v_m) = (1 - \alpha) \cdot U_V(v_v) + \alpha \cdot U_A(v_a)$.

The CPU clock cycles and bit rates needed for each variation are known as well. Let $C_e(v)$, $C_d(v)$, and $B(v)$ denote the encoding and decoding CPU clock cycles and bit rate needs of the variation v, respectively. Trivially, $C_e(v_m) = C_e(v_v) + C_e(v_a)$, $C_d(v_m) = C_d(v_v) + C_d(v_a)$, and $B(v_m) = B(v_v) + B(v_a)$. Furthermore, the CPU usage and the total bit rate of the processed streams are limited on the server. Let L_{C_e}, L_{C_d}, and L_B denote the maximum values of these resources. Let A denote the set of movie variations that satisfy the resource constraints, that is, they fall below the resource limits. These points are called appropriate: $v_m \in A \Leftrightarrow C_e(v_m) \leq L_{C_e}$, $C_d(v_m) \leq L_{C_d}$, and $B(v_m) \leq L_B$.

Our aim is to select a video and an audio variation that the AE is able to produce (5) and that each of the target features of the multimedia stream satisfies the client request (9). The CPU requirements of the server and the client have to be considered, the bit rate constraints have to be fulfilled (6), (7), and (8) and the utility of the multimedia stream resulting from their combination has to be maximized (4).

Input:

Client request: A_f for $\forall f \in F_m$.

Variations on the server (AE specific): V_v, V_a.

Limits on bandwidth and CPU usage: L_B, L_{C_e}, L_{C_d}.

Output:

Movie variation $v_m = (v_v, v_a)$.

$$\text{Maximize} \quad U(v_m) = (1 - \alpha) \cdot U_V(v_v) + \alpha \cdot U_A(v_a) \tag{4}$$

$$\text{subject to} \quad v_v \in V_v, \quad v_a \in V_a \tag{5}$$

$$C_e(v_m) = C_e(v_v) + C_e(v_a) \leq L_{C_e} \tag{6}$$

$$C_d(v_m) = C_d(v_v) + C_d(v_a) \leq L_{C_d} \tag{7}$$

$$B(v_m) = B(v_v) + B(v_a) \leq L_B \tag{8}$$

$$\begin{aligned} v_v|_f \in A_f, &\qquad \forall f \in F_v; \\ v_a|_f \in A_f, &\qquad \forall f \in F_a. \end{aligned} \tag{9}$$

It can be assumed for most of the features that the resource needs as well as the utility are monotonically increasing while the value of a feature is increasing and the other feature values remain unchanged: $(v_1|_f \geq v_2|_f) \Rightarrow U(v_1) \geq U(v_2)$,

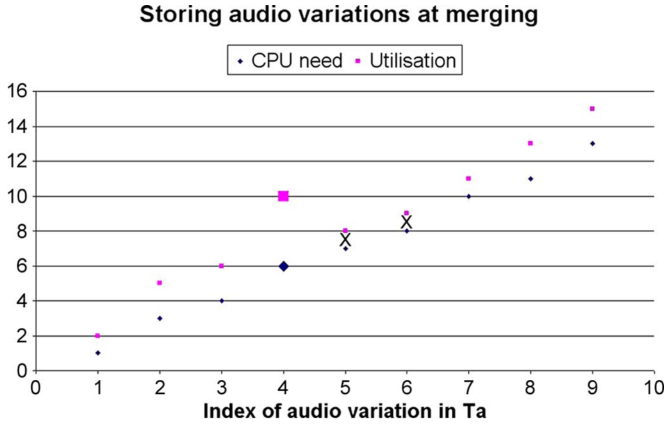


Fig. 4. Elements in T_a. The audio variation with bold symbols represents a newly inserted element. The variations with crossed symbols have to be omitted because their utilities are worse while their CPU needs are higher.

$C_e(v_1) \geq C_e(v_2)$, $C_d(v_1) \geq C_d(v_2)$, and $B(v_1) \geq B(v_2)$. This is usually true for each video and audio parameter except the video and audio codec type.

VI. Algorithms to Solve the Problem

Reference [28] gives an overview of the adaptation decision taking process. It recommends the total enumeration for the case when the possible feature values are discrete. However, more efficient algorithms are possible due to the special characteristics of the problem. In this section, we introduce the algorithms we have implemented and tested.

A. All Combinations

This approach checks all combinations of the audio and video variations to find the optimum one. This algorithm was implemented in order to validate the results of the other algorithms. The time complexity of the algorithm is $T = O(M \cdot N)$.

B. Merging Video and Audio Variations

This algorithm proceeds with video variations according to the increasing order of bandwidth demand while the audio variations are processed in decreasing order. The algorithm is looking for the best utility by generating the combination of the current video variation with the audio variation of the highest utility among those whose resource needs are less than the available resources minus the video resource need.

The algorithm can be efficiently used if the number of different resources is at most two. For this reason, we apply only two resource constraints in the implementation, namely the limits on the bandwidth and encoding CPU needs. The algorithm manages a subset of audio variations (denoted by T_a) at each step which can participate in an optimum combination with the still unprocessed video variations. The variations are ordered in T_a according to their CPU need (Fig. 4).

This method can be used for finding the minimum of a nonlinear global optimization problem which is separable into two groups, that is, the profit (utility) function and the constraints can be written as weighted sums of two variables or variable groups. In our case, the features of the audio and video variations form the two groups.

Put video variations v_v and audio variations v_a into lists $L_v()$ and $L_a()$, respectively. Order $L_v()$ and $L_a()$ according to decreasing and increasing bandwidth needs of the variations, respectively. Create empty T_a. The algorithm works as follows.

```
$j \leftarrow 1$        //index of audio variations
**for** $i \leftarrow 1, \ldots, M$ **do**        //index of video variations
    $v_v \leftarrow L_v(i)$
    $B_a \leftarrow L_B - B(v_v), C_a \leftarrow L_{C_e} - C_e(v_v)$
    $bInserted \leftarrow true$        // audio variation is inserted
    **while** $j <= N$ **and** $bInserted$ **do**
        $v_a \leftarrow L_a(j)$
        **if** $B(v_a) > B_a$ **then** $bInserted \leftarrow false$
        **else**
            Insert $v_a$ into $T_a(j), j \leftarrow j + 1$
            $v'_a \leftarrow$ Predecessor of $v_a$ in $T_a()$.
            **if** $U_a(v'_a) > U_a(v_a)$ **then** Delete $v_a$ from $T_a()$
            **else** $bDeleted \leftarrow true$
            **while** $bDeleted$ **do**
                $v'_a \leftarrow$ Successor of $v_a$ in $T_a()$.
                **if** $U_a(v'_a) < U_a(v_a)$
                **then** Delete $v'_a$ from $T_a()$
                    $bDeleted \leftarrow true$        // audio variation is deleted
    Get $v_a$ from $T_a()$ whose CPU need is highest below $C_a$.
    **if** $U(v_v, v_a) > maxU$
    **then** $v_m \leftarrow (v_v, v_a), maxU \leftarrow U(v_v, v_a)$
```

For efficiency, the ordered list of candidate audio variations is stored in a so called red-black tree, which is a special balanced tree, where look-up, insertion, and deletion can be done in $O(log(n))$ time (n is the number of nodes in the tree). In this case, the time complexity of the algorithm is $T = O(M \cdot logM + N \cdot logN)$. This can be reduced to $O((M + N)logN)$ if the video variations are ordered in advance according to their bandwidth needs.

The above algorithm is developed for the case when the number of modalities (n_m) is two. In the truly cross-modal case when the number of modalities is more than two, each combination of the possible values of $n_m - 2$ modalities have to be generated and for each combination the above algorithm has to be run with the remaining two modalities under reduced resource limits.

TABLE III
UTILITIES CALCULATED BY THE OPTIMIZATION ALGORITHMS AS PERCENTAGES OF THE OPTIMUM

	All combinations	*Merging*	*Border scan*	*Hill climbing*	*Improved hill climbing*	*Size reduction*
Average relative utility	100%	100%	100%	99.0%	99.8%	94.7%
Minimum relative utility	100%	100%	100%	85.0%	98.7%	76.1%

C. Border Scan

This algorithm exploits the monotonicity of the utility and resource needs in the feature values. There are several methods for optimization where the goal function as well as the constraints are monotonic [29]. We applied another method to find the optimum in the monotonic case: the points are enumerated and compared with each other that are located at the surface of the resource constraints in the joint feature space of all modalities.

In the feature space of d dimensions, the resource constraints determine a surface (border) of $d-1$ dimensions, that separates the movies that comply with each of the resource constraints from the movies that violate any of them. From the monotony of the resource needs it follows that all points below the surface comply with the resource constraints and all points above it do not. From the monotony of the utility it follows that the optimum point is located directly below the border; that is, increasing any of its parameters to the next higher value, if any, results in a variation that needs too much resources. (We call these points the *border points*). As a consequence, it is enough to examine the appropriate points along the border when we search the one with the highest utility. Unfortunately, there is no guarantee on the monotony of the utility along the border, so we have to search the optimum solution at the whole surface of the border.

First, the algorithm looks for a single border point moving in the direction of one selected feature denoted by f_1. Then the algorithm considers the further monotonic features f_i, $i = 2, \ldots n$, one after the other. For each feature, the algorithm generates the border points restricted to the space of the first i features. Let the set of these points be denoted by B_i. For each border point in B_{i-1}, the value of the current feature f_i is gradually increased while the value of feature f_1 is decreased if necessary in order to create border points in B_i. After considering each feature, we select the border point with the highest utility as optimum.

The detailed description of the algorithm can be found below. For simplicity, we assume that each feature is monotonic. Otherwise, the algorithm has to be repeated for the different values of the nonmonotonic feature such as codec type or for each combination of values of nonmonotonic features if there are more nonmonotonic features.

```
for each $f \in F_m$ do $g_{f,i_f} \leftarrow g_{f,1}$
// Going to the border
while $(g_{f_1,i_1+1}, g_{f_2,1}, \ldots, g_{f_n,1}) \in A$ do
    $i_1 \leftarrow i_1 + 1$
$t \leftarrow 1$        // index of border points
$b_t \leftarrow (g_{f_1,i_1}, g_{f_2,1}, \ldots, g_{f_n,1})$
$t_1 \leftarrow 1$        // number of border points
for $k \leftarrow 2 \ldots n$ do        // features
    for $j = 1, \ldots, t_{k-1}$ do        // stored border points
        $(g_{f_1,i_{f_1}}, \ldots, g_{f_{k-1},i_{k-1}}, g_{f_k,1}, \ldots, g_{f_n,1}) \leftarrow b_j$
// Increasing the current feature.
        for $i_k = 2, \ldots, l(f_k)$ do        // feature values
            $i'_{f_1} \leftarrow i_{f_1}$
// Decreasing the value of the first feature
            while $(g_{f_1,i'_{f_1}}, \ldots, g_{f_k,i_k}, g_{f_{k+1},1}, \ldots, g_{f_n,1}) \notin A$
                    and $i'_{f_1} > 1$ do
                $i'_{f_1} \leftarrow i'_{f_1} - 1$
            if $(g_{f_1,i'_{f_1}}, \ldots, g_{f_k,i_k}, g_{f_{k+1},1}, \ldots, g_{f_n,1}) \in A$
            then $t \leftarrow t + 1$
                $b_t \leftarrow$
                $(g_{f_1,i'_{f_1}}, \ldots, g_{f_k,i_k}, g_{f_{k+1},1}, \ldots, g_{f_n,1})$
    $t_k \leftarrow t$
Find $j = j'$ for which $U(b_j)$ is maximum such that $1 \leq j \leq t$
The selected variation is $b_{j'}$
```

The time complexity of our border scan algorithm is $T = O(M \cdot N / \min(l(f_1), l(f_n)))$.

D. Hill Climbing

Due to the monotonicity in the resource needs and utility, we could use a heuristic search method, namely steepest-ascent hill climbing [30] as well, and we found it as an efficient approach for the real-time application at hand. We start with the worst variation. In each iteration step, we increase the value of the monotonic feature where the utility increase is the highest and the improved variation still satisfies the resource constraints. This algorithm does not necessarily find the optimum because it may run into a local maximum at the border defined by the constraints but it is clearly the fastest algorithm in practical cases. The time complexity of the algorithm is: $O(\sum_i l(f_i))$.

We tried out several modifications in order to avoid local maxima and to improve the goodness of this simple algorithm. We could achieve significant improvement (see Table III) by starting the algorithm from different initial points and then selecting the best variation from the results of different runs. We selected as many additional initial points in our implementation as the number of the stream features.

E. Performance Results

We implemented the above algorithms and ran them on real multimedia stream data. In an earlier paper [13], we examined the implementations. We used a screen shot from the Batman film for the tests. The quality of the original video was characterized by the following parameters: spatial resolution: 720 × 576, frame rate: 25 fps, video quantization: 1, audio sample

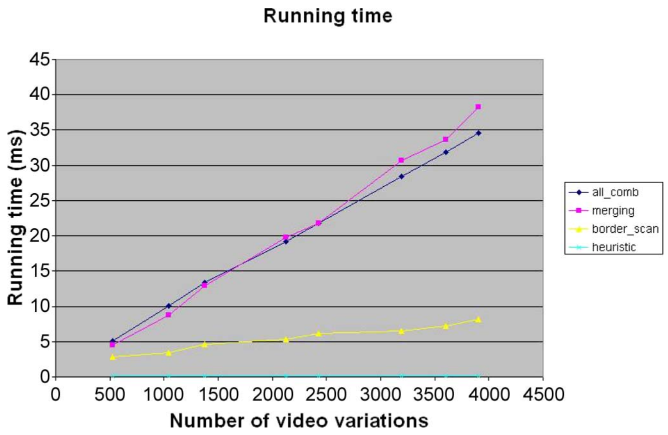


Fig. 5. Running times of the algorithms as a function of the number of video variations.

rate: 48 kHz, encoding audio bit rate: 160 kb/s, number of audio channels: 2. The parameters of the available variations were as follows: there are 5 different possible values for the spatial resolution (720 × 576, 704 × 576, 640 × 480, 320 × 200, 175 × 144), 25 for the frame rate (1–25 fps), 31 for the video quantization (1–31), 3 for the audio sample rate (48 kHz, 44 kHz, 22 kHz), 2 for the number of audio channels (mono, stereo) and the encoding audio bit rate had two different values for each combination of the audio sample rate and the number of audio channels. In this case, the total number of different video and audio variations was 3096 and 12, respectively. We generated different optimization tasks for this video by varying the high level parameters and resource constraints. We extended the tests by including the improved version of the hill climbing algorithm. We also examined the method when only the size was reduced until the result did not fit into the resource constraints and then its utility was calculated. This later method is included in order to show how much improvement can be achieved by the utility based adaptation relative to a traditional method which neglects the utility aspects (Table III). Each recommended algorithm was much better than reducing the size only. Clearly, heuristic search (hill climbing) was the fastest but it does not always find the exact optimum. We observed that it is more likely to fail finding the optimum if the resource limits are low. Its goodness could be improved by repeating it from different initial points. Generating all combinations was not too inefficient because the number of audio variations was small in these experiments. Merging was the slowest but its running time can be highly reduced if sorting is done in advance before client requests arrive.

We tested the algorithms on inputs with different numbers of video variations as well. Fig. 5 shows the average running times of the implemented algorithms as a function of the number of video variations. The test were running on a 1.1-GHz processor with 256-MB memory. In case of generating all combinations, we can see that the running time depends linearly on the number of the video variations if the number of audio variations is fixed. The running time of merging increases a little bit faster (its running time contains $O(M \cdot \log M)$). We remark that the running time of generating all combinations could increase faster than the merging method if the audio variations would also increase.

VII. Conclusion

We presented a generic A/V utility model for multimedia content adaptation that is able to consider the user and the usage environment as well as different genres. For efficient and personalized parameter setting, we introduced a recommender-based approach that configures the model to the user's individual utility notion based on intuitive rules, users' judgements, demographic features, and favorite content types. Finding video and audio stream variations that optimize the media stream's utility (or the media experience) for the user based on the proposed model under given resource constraints, represents a complex optimization problem in the multimedia area. We presented and implemented four algorithms to find optimal video and audio variations for multimedia content adaptation. We found the simple heuristic hill-climbing optimization method to be the most efficient. However, this algorithm may fail to find the optimum, so it has to be used with care and potentially has to be improved. The merging method is recommended especially when the utility function is nonmonotonic and preparation (sorting) can be done before client requests arrive. Border scan is efficient in the monotonic case if the hill climbing approach fails.

Applying the presented approach to an adaptive multimedia framework yields a better multimedia experience for the client. Further experimental work will be performed to appropriately fit the high level parameters to different usage environments and genres. To that end, the recommender approach will be refined by analyzing and taking into account the feedback history created by a larger test community.

References

[1] A. Vetro, C. Christopoulos, and T. Ebrahami, Eds., *IEEE Signal Process. Mag., Special Issue on Universal Multimedia Access*, vol. 20, no. 2, Mar. 2003.

[2] D. Jannach, K. Leopold, C. Timmerer, and H. Hellwagner, "A knowledge-based framework for multimedia adaptation," *Appl. Intell.*, vol. 24, no. 2, pp. 109–125, Apr. 2006.

[3] F. López, J. M. Martínez, and V. Valdés, "Multimedia content adaptation within the CAIN framework via constraints satisfaction and optimization," presented at the Proc. 4th Int. Workshop on Adaptive Multimedia Retrieval (AMR), Geneva, Switzerland, Jul. 2006.

[4] F. Pereira and I. Burnett, "Universal multimedia experiences for tomorrow," *IEEE Signal Process. Mag.*, vol. 20, no. 2, pp. 63–73, Mar. 2003.

[5] H. Knoche, J. D. McCarthy, and M. A. Sasse, "Can small be beautiful? Assessing image resolution requirements for mobile TV," in *Proc. ACM Multimedia*, Nov. 2005, pp. 829–838.

[6] A. Vetro and C. Timmerer, "Digital item adaptation: Overview of standardization and research activities," *IEEE Trans. Multimedia*, vol. 7, no. 3, pp. 418–426, Jun. 2005.

[7] T. C. Thang, Y. J. Jung, and Y. M. Ro, "Modality conversion for QoS management in universal multimedia access," in *Proc. IEE—Vis., Image, Signal Process.*, Jun. 2005, vol. 152, pp. 374–384.

[8] S. Winkler, *Digital Video Quality: Vision Models and Metrics*. New York: Wiley, Mar. 2005.

[9] T. C. Thang, Y. J. Jung, and Y. M. Ro, "Semantic quality for content-aware video adaptation," in *Proc. IEEE MMSP'2005*, Oct. 2005, pp. 1–4.

[10] S. Winkler and C. Faller, "Audiovisual quality evaluation of low-bit rate video," in *Proc. SPIE/IS&T Human Vis. Electron. Imag.*, Jan. 2005, vol. 5666, pp. 139–148.

[11] D. S. Hands, "A basic multimedia quality model," *IEEE Trans. Multimedia*, vol. 6, no. 6, pp. 806–816, Dec. 2004.

[12] M. Prangl, H. Hellwagner, and T. Szkaliczki, "A semantic-based multimodal utility approach for multimedia adaptation," in *Proc. 7th Int. Workshop Image Anal. Multimedia Services (WIAMIS)*, Apr. 2006, pp. 67–70.

[13] M. Prangl, H. Hellwagner, and T. Szkaliczki, "Fast adaptation decision taking for cross-modal multimedia content adaptation," in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME)*, Jul. 2006, pp. 137–140.

[14] A. Cavallaro, O. Steiger, and T. Ebrahimi, "Semantic video analysis for adaptive content delivery and automatic description," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 10, pp. 1200–1209, Oct. 2005.

[15] P. Salembier and J. R. Smith, "MPEG-7 multimedia description schemes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 6, pp. 748–759, Jun. 2001.

[16] S. Winkler and C. Faller, "Maximizing audiovisual quality at low bit rates," presented at the Workshop Video Process. Quality Metrics Consum. Electron., Jan. 2005.

[17] B. Feiten, I. Wolf, E. Oh, J. Seo, and H.-K. Kim, "Audio adaptation according to usage environment and perceptual quality metrics," *IEEE Trans. Multimedia*, vol. 7, no. 3, pp. 446–453, Jun. 2005.

[18] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Trans. Broadcasting*, vol. 50, no. 3, pp. 312–322, Sep. 2004.

[19] *Methodology for the Subjective Assessment of the Quality of Television Pictures*, ITU-R BT500-11, 2002.

[20] M. Montaner, B. Lopez, and J. De la Rose, "A taxonomy of recommender agents on the internet," *Artif. Intell. Rev.*, vol. 19, pp. 285–330, 2003.

[21] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.

[22] A. I. Schein, A. Popescul, and L. H. Ungar, "Methods and metrics for cold-start recommendations," in *Proc. 25th Ann. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval*, 2002, pp. 253–260.

[23] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[24] A. Felfernig and A. Kiener, "Knowledge-based interactive selling of financial services using FSAdvisor," in *Proc. 17th Innovative Appl. Artif. Intell. Conf. (IAAI'05)*, 2005, pp. 1475–1482.

[25] T. Schiex, H. Fargier, and G. Verfaille, "Valued constraint satisfaction problems: Hard and easy problems," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI'95)*, 1995, pp. 631–639.

[26] K. McCarthy, L. McGinty, B. Smyth, and J. Reilly, "A live-user evaluation of incremental dynamic critiquing," in *Proc. 6th Int. Conf. Case-Based Reas.*, 2005, vol. 3620, Springer LNCS, pp. 339–352.

[27] M. Grĉar, "User profiling: Collaborative filtering," in *Proc. Conf. Data Mining Data Warehouses (SiKDD)*, Ljubljana, Slovenia, Oct. 2004, pp. 75–78.

[28] D. Mukherjee, E. Delfosse, J.-G. Kim, and Y. Wang, "Optimal adaptation decision-taking for terminal and network quality of service," *IEEE Trans. Multimedia*, vol. 7, no. 3, pp. 454–462, Jun. 2005.

[29] H. Tuy, "Monotonic optimization: Problems and solution approaches," *SIAM J. Optimiz.*, vol. 11, no. 2, pp. 464–494, 2000.

[30] E. Rich and K. Knight, *Artifical Intelligence*. New York: MacGraw-Hill, 1991.

Martin Prangl (S'07) received the B.S. and M.S. degrees in telematics from the Technical University of Graz, Graz, Austria.

He joined Klagenfurt University, Klagenfurt, Austria, in 2004 and is currently a University Assistant at the Department of Information Technology (ITEC) where he is pursuing the Ph.D. degree in multimedia adaptation. He has been working on quality and utility modelling of multimedia content and live television content adaptation. Other research interests include medical imaging, digital video broadcasting (DVB), the transport of multimedia content, multimedia adaptation in constrained and streaming environments, distributed multimedia adaptation, radio communication, and electronics.

Tibor Szkaliczki (M'06) received the M.S. and Ph.D. degrees in electrical engineering from the Technical University of Budapest, Budapest, Hungary, in 1992 and 1998, respectively.

He joined the Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, Hungary, in 1997 and is currently a Research Fellow at the eLearning Department. He received research scholarship at Klagenfurt University, Klagenfurt, Austria, for five months in 2005. His research interests include distributed multimedia adaptation, multimedia adaptation in constrained and streaming environments, VLSI routing, and combinatorial optimization.

Hermann Hellwagner (S'85–M'95) received the M.S. and Ph.D. degrees in informatics from the University of Linz, Linz, Austria, in 1983 and 1988, respectively.

He has been a Full Professor of informatics in the Department of Information Technology (ITEC), Klagenfurt University, Austria, for eight years. His current research areas are distributed multimedia systems, multimedia communications, Internet QoS, and MPEG-21. He has received many research grants from national (Austria, Germany) and European funding agencies as well as from industry. He is the editor of several books and has published widely on parallel computer architecture and parallel programming and, more recently, on multimedia communications and adaptation. He is the Head of the Austrian delegation to the Moving Picture Experts Group (MPEG-ISO/IEC JTC1/SC29/WG11).

Dr. Hellwagner has organized several international conferences and workshops. He is a member of the German Informatics Society (GI) and the Austrian Computer Society (OCG).