Multimedia Systems manuscript No.
(will be inserted by the editor)

Benjamin Köhncke · Wolf-Tilo Balke

Preference-Driven Personalization for Flexible Digital Item Adaptation

Abstract The delivery of multimedia content often needs the adaptation of the content in order to satisfy user constraints. With the Digital Item Adaptation part, the MPEG-21 standard already defines a useful frame-work to handle this task. However, in modern service-oriented architectures the functionality of adaptation is split over several services. Hence, the central instantiation of a suitable service chain needs to tackle a complex multi-objective optimization problem. In this problem between content choice and possible adaptations the current preference model in the MPEG-7/21 standard still lacks expressiveness. In the course of this paper we demonstrate this shortcoming and how the integration of more powerful models can ease the instantiation problem. Furthermore we explain how to efficiently evaluate preference trade-offs by evaluating skyline queries as currently investigated in the field of information systems. As a running example we use preference-based content adaptation in a typical media streaming application with Web services as basic modules. The contribution of our framework is to enable a central coordinator to instantiate an executable service composition chain by integrating all needed Web services to adapt the multimedia content in the best possible fashion in the sense of Pareto optimality.

Keywords Multimedia Systems · Digital Item Adaptation · Personalization · Skyline Queries · Preference Modeling

This work was supported in part by the German Research Foundation (DFG) within the Emmy-Noether Program of Exellence.

Wolf-Tilo Balke
L3S Research Center at the University of Hannover, Hannover, Germany
Tel.: +49-511-762-17712
Fax: +49-511-762-17779
E-mail: balke@l3s.de

Benjamin Köhncke
L3S Research Center at the University of Hannover, Hannover, Germany
E-mail: koehncke@l3s.de

1 Introduction

Delivering multimedia content over a plethora of devices in a personalized fashion puts great demands on the selection and adaptation of content [24]. There are two basic approaches: either a server side adaptation, where all adapted versions of the appropriate content are then broadcasted in a stream, or a client side adaptation where powerful decoders have to be available, to perform an on-the-fly adaptation. Building on service-oriented adaptation our approach goes beyond these basic models and provides an individual service chain for each user where the adaptation is done on-the-fly. For every individual user a specific workflow has to be created respecting the individual user's content preferences, general content semantics and network constraints, as well as terminal capabilities.

This is because today, instead of building complex monolithic systems the flexible composition of services into suitable workflows for media handling is an important goal for system integration and reusability of components (for an overview see [17]). Such workflows have to be planned and their execution closely monitored, sometimes needing the seamless integration of replacements for failed services. To facilitate this, the instantiation of the service chain needs to solve a multi-objective optimization problem considering possible services (as have been discovered, cf. [1]), the content currently available and the author's, user's and client device's constraints on the adaptation.

After having addressed the basic problem in [11], in this paper we discuss how to adapt content in a personalized fashion by integrating MPEG-7/21 metadata with a complex preference framework, namely a model on qualitative partial order preferences (cf. [6], [9]) and show how to evaluate the complex result meshes given by the respective product order under the concept of Pareto optimality.

The benefit is twofold: on one hand integrating a more powerful model enables higher expressiveness of user preferences; on the other hand composition engines and ser-

vices along the workflow chain can directly use the augmented metadata for choosing adaptation strategies more intelligently. Every service adapts the metadata according to the transformation of the content it performed and removes obsolete preference information. This also enables a more efficient matching of constraints for those services later in a service composition chain.
As a running example application throughout this paper we will use media streaming that flexibly adapts some multimedia content to the terminal capabilities of the user. Thus, users get the best possible quality with respect to their respective terminal capabilities. Our prototypical implementation uses Web services as basic modules to build multimedia applications. The description of the complex data types is provided by an MPEG-7/21 description attached to the media data. Each individual Web service evaluates the MPEG-7/21 description and adapts the multimedia material to the special needs of the user and the client device.
To coordinate the composition we currently use a dedicated central Web service instance that takes over responsibility for selecting the best available content and a suitable workflow by means of the technical profile and user preferences [1]. This *service instantiation and monitoring service (SIAM)* does also monitor the workflow, but it does not in detail decide in what way each Web service should adapt the multimedia content. Every individual Web service (e.g., merging or transcoding services) can use its own heuristics about how to adapt the selected multimedia content best with respect to the specified target output for the end-user. In addition to the purely technical information, the services also use important description schemes from the MPEG-7/21 standard like e.g., transcoding hints [23].
The rest of the paper is structured as follows: In section 2 we will give an overview about the user specific metadata descriptions offered by MPEG-7/21. A short summary about what has already been done in the field of personalized media adaptation is given in the related work part in section 3. Then we introduce a more expressive preference model in section 4 and present the basic architecture of our prototypical system. In section 5 we describe how to actually evaluate the complex trade-off mesh of a Pareto preference graph and the prototypical implementation of a media streaming test case is discussed in section 6. Finally, we close the paper with our conclusions and some future research directions.

2 Metadata Descriptions in MPEG-7/21

The MPEG-7/21 *Usage Environment* is a part of the MPEG-21 Digital Item Adaptation (DIA) architecture (Part 7, ISO/IEC 21000-7) and offers several opportunities to define user specific information. It includes the *User Interaction Tools* from MPEG-7 [23] and also the description elements for the device capabilities of the user (in the *Terminal Capabilities* description scheme). Figure 1 shows the typical interaction of a content adaptation engine with a media database and the end user, respectively his/her client device. Media data generally

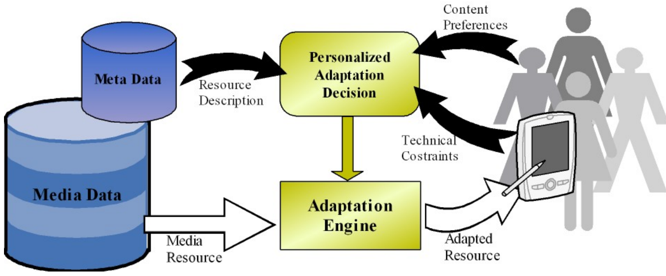

Fig. 1 General Digital Item Adaptation

is retrieved by matching metadata descriptions. To perform a content selection the personalized adaptation decision engine matches the resource descriptions from a metadata directory and the user-provided information. This information consists of the actual user query, more general content preferences of the user, and technical constraints that provide data about the technical device of the user. After determining the optimal adaptation, the decision is handed on to the adaptation engine. The adaptation engine then retrieves the media resource from the media database, adapts it accordingly and delivers the adapted resource to the user. Basically an adaptation works as follows:

1. The decision engine gets a request with content preferences and technical constraints
2. If the requested media is available and all constraints are satisfied the respective multimedia data is directly delivered
3. If the requested media is available, but some constraints are not yet satisfied
   3.1. If the multimedia data can be adapted to meet all constraints, adapt media accordingly and deliver
   3.2. If not, choose the highest ranking preference(s) to adapt to other formats, resolutions, etc. or to choose new content

However, this adaptation scheme cannot work with more complex trade-offs, and neither can it decide intelligently when to choose alternative adaptation options that (possibly if taken together) would make more sense than simply the highest rated preference. Let us take a closer look on what MPEG-7/21 can actually express in user preferences.
The *User Interaction Tools* (UIT) offer several possibilities to describe content preferences of the user. In contrast to the other description schemes defined in MPEG-7 the UIT are bound to the user and not to the multimedia material: the *User Preferences* are used to express specific preferences of the user in the selection of

multimedia material and consist of several description schemes:

- A set of *Filtering and Search Preferences* describes individual user wishes with respect to the filtering and searching of multimedia data. Each element is decomposed into:
  - *Filtering and Search Preference* elements on a lower level to allow for preference hierarchies.
  - *Classification Preference* elements to describe preferences regarding to attributes, such as language, production format or country of origin.
  - *Creation Preference* elements to describe the preferred creation of content, such as title, creator or creation date.
  - *Source Preference* elements where preferred repositories to retrieve multimedia material from (e.g., media servers or digital libraries) can be defined.
  - *Preference Condition* elements where the user can constrain the applicability of specific filtering and search preferences (for example time and date).
- A set of *Browsing Preferences* elements describe the user's wishes regarding multimedia content navigation and browsing. A typical example is a *Summary Preference* describing the preferred type of content summaries.

If the user got more than one preference, a weighting factor, called *Preference Value*, can be specified to express the relative importance of a preference. The preference value is a simple numerical value ranging from -100 to 100. The value indicates the degree of users preference or non-preference. The zero value indicates that the user is neutral in terms of preference versus dislike. A default, positive, value of 10 corresponds to a nominal preference. By choosing a negative weighting the user can express negative preferences or dislikes.

Example: The following XML code snippet shows a complex preference in MPEG-21 notation for user 'Kim', who prefers 'Action' movies starring 'Matt Damon' over the actors 'Arnold Schwarzenegger' and 'Brad Pitt' and on the whole movies that are in German language over movies in English language using appropriate *Filtering and Search Preference* elements.

```
<UserPreference>
  <UserIdentifier userName="Kim"/>
  <UsagePreference>
    <FilteringAndSearchPreferences>
      <ClassificationPreference>
        <Genre>Action</Genre>
        <Language preferenceValue="90">german</Language>
        <Language preferenceValue="70">english</Language>
      </ClassificationPreference>
      <CreationPreference>
        <Actor preferenceValue="63">Damon</Actor>
        <Actor preferenceValue="50">Schwarzenegger</Actor>
        <Actor preferenceValue="27">Pitt</Actor>
      </CreationPreference>
    </FilteringAndSearchPreference>
  </UsagePreference>
</UserPreference>
```

To decide between preferences all the preference values within each attribute are compared separately: the higher the value, the more important the respective preference. However, the MPEG-7/21 standard just defines the syntax and semantics of the user preference description scheme, but not the extraction method of the preference value, cf. [12]. That means all users have to assign preference values *manually.* But can semantically incomparable attributes (like preferences on actors and language settings) be compared in a quantitative way? It hardly makes sense to state something like: 'I prefer Matt Damon movies to movies in German'. Moreover, for the user it is entirely unintuitive, what an individual preference value (like 63 or 50) actually means. Other authors, e.g. [27], have proposed to use even more complex utility functions, but after all still rely on quite unintuitive quantitative preference frameworks.

In the example above we have two preferences on different attributes: language and actors, but the attributes are basically incomparable. Thus, some combinations for media objects become also incomparable and cannot be ranked in a total order. This characteristic leads directly to the concept of Pareto optimality: the Pareto set (often also called efficient frontier) consist of all non-dominated objects, i.e. for each object no other object in the set has better or at least equal attribute values with respect to all attributes. Consider for instance an English Matt Damon movie and a German Arnold Schwarzenegger movie. Both are incomparable, because one is better with respect to the language preference, whereas the other is better with respect to the actor preference. However, both options dominate an English Brad Pitt movie, which accordingly would not be part of the Pareto set. The use of the Pareto semantics is also advocated in [14] providing a decision taking framework where hard- and soft-constraints are represented as variables as input for the optimization problem.

A simple matching of preference values rarely leads to an effective trade-off management. This is because preferences generally distinguish between hard and soft constraints. Hard constraints have to be adhered to, no matter how (un-)important the respective preference is. Consider transcoding hints, where the original author of the multimedia material can define how the properties of the multimedia content can be changed without compromising the content's semantics. For instance, it could be stated that the resolution of a movie can only be reduced up to 50% of the original resolution. A further reduction simply does not make sense; even if it is exactly the content a user requested by expressing content preferences with high preference values. On the other hand a user might express a preference for best possible resolution. Such a preference can be considered as a soft constraint that can be relaxed, if necessary.

Generally speaking two hard boundaries always constrain service compositions (cf. figure 2): the device capabilities form upper boundaries for the capabilities of the terminal, like for example the maximum possible resolution. The transcoding hints form lower boundaries for the quality of multimedia material, like for example the minimum possible resolution. If the device capabilities and transcoding hints do not overlap a sensible media adaptation without changing the modality is not possible (cf. figure 2).

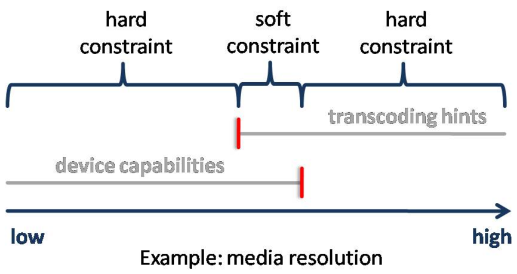

Fig. 2 Boundaries for Media Adaptation

Basically for adaptation of the multimedia material our SIAM service uses the InfoPyramid approach [13] to offer multimedia content in alternative fidelities and/or modalities. In case of impossible adaptations the user's content preferences are used to choose alternative media and thus realize a truly cooperative retrieval behavior.

3 Related Work

The basic usefulness of the user preferences description scheme for automatic content adaptation has already been shown to some degree by a variety of approaches. [25] provides an useful overview of current digital item adaptation techniques in MPEG-21 and how they can be utilized by multimedia applications. In particular, the authors give detailed insights about *usage environment* descriptions which includes, among other things, *terminal capabilities* and *network characteristics*. Furthermore ongoing standardization activities related to DIA, as well as several emerging research topics and open issues are discussed.
The semantic information a multimedia document conveys can be presented within different levels of abstraction. Recently [19] described how to include semantic descriptions using semantic Web technology. But since today semantic Web languages still lack the structural advantages of the XML-based approach and current multimedia document annotation is already given by other legacy standards, a combining of the existing standards seems to be the most promising path for multimedia document description. MPEG-7/21 as one of the international key standards in this area is defined by a XML schema and currently no commonly accepted mapping exists from the XML schema definitions to RDF or OWL [19]. Although a further monitoring of the development of semantic Web media descriptions is necessary, currently we still rely on XML annotations.
In terms of adaptation frameworks [20] also uses the simple XML model to design a video personalization and summarization system in heterogeneous usage environments. The design framework results in a three-tier architecture of server, middleware and client. Personalization and adaptation engines, which select, adapt and deliver summarized media files to the user, are included in the middleware layer. For each decision if a video shot should be included in the personalized video summary or not, the preference scores from the MPEG-7/21 standard are taken. However, unlike in our work presented here in all cases these preferences are static and the appropriate adaptation is always assumed to be possible.
In contrast, the work in [21] goes beyond mere static adaptation and even provides some novel ontology-based methodologies to open up the MPEG-7/21 usage environment for enriching user preferences by more complex semantics as expressed by domain ontologies. In doing so, the approach supports the complete functionality offered by the MPEG-7 semantic description scheme for multimedia content descriptions and respects all the MPEG-7/21 conventions. It is based on the Web ontology language OWL and has been prototypically implemented in the DS-MIRF framework. Moreover, in [22] the system is extended by a Query language for MPEG-7 descriptions. Allowing the integration of domain knowledge into semantic MPEG-7 metadata an automatic relaxation of over-specified constraints is thus enabled. However, the actual integration of specific user preferences is not yet supported in this framework and the task of personalization still remains a challenging issue.
In a similar fashion [15] explores diverse algorithms and methodologies for an ontology-based approach, but here the focus is on exchanging knowledge between users. The semantic relations in the ontology are defined in MPEG-7 and can be visualized using RDF. An agglomerative clustering approach is used to extract preferences based on user history and a fuzzified version of this knowledge can be exchanged.
To exploit also specific user preferences and therefore improve the user's interaction experience and understanding of the adaptation process the integration of a more expressive preference model is needed. An important step to derive and maintain user preferences in a meaningful profile is the elicitation step. There exists a large body of work in user modeling (see e.g., [10], [5] for a basic overview) relying on different techniques to derive a profile. Especially suitable group profiles as compromises or a basis for multicast techniques may be helpful in solving some of the demanding QoS issues involved in media streaming. For instance, in [12] a use case of an electronic program guide is presented, where the user's interactive behavior with video programs is monitored, and the user preferences are updated accordingly. Here, the relative importance of different preferences can to some degree

even be generated automatically.
In terms of the service-oriented processing of multimedia data the MPEG-21 standard also introduces network quality of service (QoS) tools. For choosing adequate network paths the usefulness of integrating more complex user preferences has already been recognized and usually typical non-functional statistics like average service availability or network latencies are already integrated in the decision, e.g. for telephony [8]. However, the actual real-time measurement of statistical information to decide for best service paths is still difficult and beyond the scope of this paper (see e.g., [7] or [16]).

4 A More Expressive Preference Model

The existence of multiple and often conflicting user preferences demands an efficient framework to deal with them in a fair and meaningful way. The need of an effective trade-off management with complex user preferences has already been considered in other communities like e.g., in databases and information systems. Here, recent work in [6] and [9] considers preferences in a qualitative way as partial orders of preferred values that can be relaxed if the need arises. To combine multiple preferences and derive a fair compromise usually the notion of Pareto optimality is used.
In the case of content selection and adaptation the results are all possible solutions that are not dominated by other solutions with respect to all attributes specified in the media request. That means if all preferences on different attributes are considered to be of equal importance, the sub-optimal solutions are automatically removed and out of the remaining pool of possible solutions an adaptation service can pick a suitable instantiation. Of course, if no fair relaxation scheme is desired, also more discriminating combination methods (e.g. the ordering on the attributes in preference values in MPEG-7/21) can be used on qualitative partial order preferences.

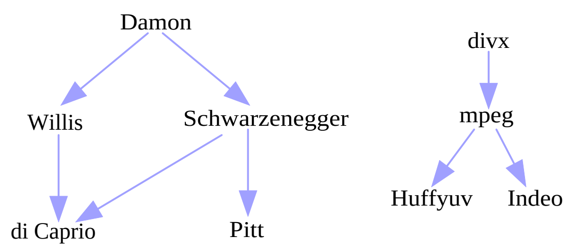

Fig. 3 Explicit Preference Graphs

Example (cont.): Consider our user Kim owns a PDA and again requests some media for streaming, but this time explicitly expresses two complex preferences: one about preferred movie actors and one about preferred codecs available of her client device (as defined by the terminal capabilities). The corresponding preference graphs might look as shown in figure 3. Our framework extends the MPEG-21 notation to express these preferences in XML, e.g., the first preference, like follows:

```
<UserPreference>
  <UserIdentifier userName="Kim"/>
  <UsagePreference>
      <Preference>
        <EXP att="actor">
          <EXPSet>
            <Value val1="Damon" val2="Willis">
            <Value val1="Damon" val2="Schwarzenegger">
            <Value val1="Willis" val2="di Caprio">
            <Value val1="Schwarzenegger" val2="di Caprio">
            <Value val1="Schwarzenegger" val2="Pitt">
          </EXPSet>
        </EXP>
      </Preference>
  </UsagePreference>
</UserPreference>
```

To combine several preferences the combination semantics has to be stated. For our example let us assume a fair relaxation scheme between the two preferences. Figure 4 shows the first three layers of the product preference graph following the Pareto semantics. The graph is already quite complex for combining only the two preferences in figure 3. Please note that due to the qualitative nature of the preferences some combinations are incomparable: if the best choice (a Matt Damon movie in divx format) should not be available or adaptable (e.g., missing a suitable transcoding service), the adaptation decision can explore several other options that are all equally preferable such as a Matt Damon movie in mpeg format or a Bruce Willis movie in divx format.

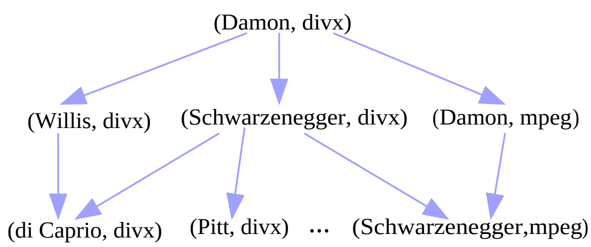

Fig. 4 Pareto Preference Graph

To express such Pareto preferences in XML we allow qualifying the combination of explicit preferences following the respective preference algebra (see [9] for details on the evaluation of complex constraints):

```
<UserPreference>
  <UserIdentifier userName="Kim"/>
  <UsagePreference>
      <Preference>
        <Pareto>
          <EXP att="actor">
            <EXPSet>
              <Value val1="Damon" val2="Willis">
              <Value val1="Damon" val2="Schwarzenegger">
              <Value val1="Willis" val2="di Caprio">
              <Value val1="Schwarzenegger" val2="di Caprio">
              <Value val1="Schwarzenegger" val2="Pitt">
            </EXPSet>
```

```
          </EXP>
          <EXP att="codec">
            <EXPSet>
              <Value val1="divx" val2="mpeg">
              <Value val1="mpeg" val2="Huffyuv">
              <Value val1="mpeg" val2="Indeo">
            </EXPSet>
          </EXP>
        </Pareto>
      </Preference>
  </UsagePreference>
</UserPreference>
```

For evaluating such an advanced trade-off management in a service-oriented environment several components are needed. Figure 5 shows the basic architecture of our preference trade-off management. The user requests media from our central SIAM service and provides constraints and rich preferences in the form of content preferences (UP) and technical capabilities (TC). The decision engine negotiates the content by retrieving all necessary metadata integrating the respective transcoding hints (TH) into the preference trade-off management.

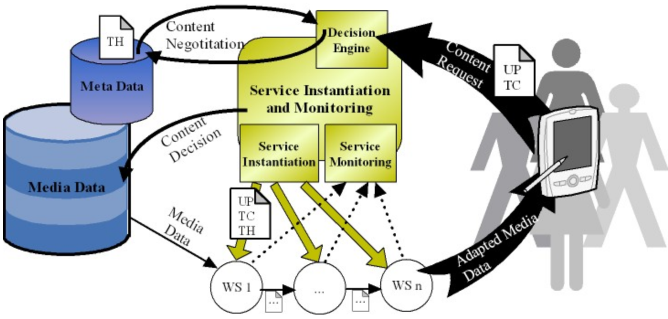

Fig. 5 Service-oriented Adaptation

The decision engine part of the SIAM service decides by means of the given constraints which Web services are basically needed to perform an adequate media delivery. Possible Web services (which are discovered by the service instantiation part) have to respect the preferences (and QoS constraints) to choose an optimal adaptation of the multimedia data for each individual user.
The actual instantiation and monitoring engines used in our SIAM architecture are not the focus of this paper and have been extensively discussed in [1]. In a nutshell, the E^2Mon (Execution and Environment Monitoring) framework allows a fast selection of new service chains and a quality and cost based service failure recovery. To find an optimal service chain a cost function is used that considers the parameters from the actual execution (success or failure) and furthermore a variety of costs depending on device capabilities or the execution environment. If the costs of already discovered services change (e.g., the expected QoS), either a provider advertises the change in a service or the environment has changed (e.g., roaming to a new network or a low battery warning). In either case an event is raised and a rediscovery would initiate the change to the cost-optimal solution, while reusing the results of already invoked services when new best service chains are derived.
E^2Mon is itself run as a Web service either on the client or (more often) on a proxy machine to reduce the communication load of the client device for Web service invocation and monitoring. Of course, such a proxy-based approach is only useful if alarms (e.g. battery low) are rare compared to external events (service failures), as otherwise communicating the alarms can lead to a high communication load.

5 Efficiently Evaluating Preference Trade-Offs

To efficiently evaluate the complex mesh of the Pareto product order induced by preferences specially adapted algorithms are needed. Due to their practical usefulness recently in the field of databases some attention has been paid to concepts for retrieving Pareto optimal sets - so-called skyline queries, see e.g., [4] or [18].
Skyline queries describe the case where all query attributes are considered to be independent and equally important. Hence, no weighting function combining individual attribute scores like in conventional in top-k retrieval, can be used. Instead, all possibly optimal objects, based on the notion of Pareto optimality, are returned to the user. An object is considered to be optimal with respect to a collection of objects and a set of preferences, if it is not dominated by any other object. An object dominates another object, if it shows more or at least equally preferred attribute values as this object with respect to all attributes and is strictly preferred in at least one attribute. However, skyline queries have been defined with two important constraints: the data has to be totally ordered for each attribute and for easy comparisons attributes have to be defined over numerical domains.
Within skyline frameworks users are also offered the possibility to declare several hard constraints on attributes, as given e.g., by transcoding hints or terminal capabilities. This is usually facilitated as a simple selection condition for filtering [4]. But for the use in adaptation frameworks hard constraints have to be further distinguished. This is because some hard constraints can still be met by adapting content (like a codec or resolution constraint), whereas others can never be amended by altering a media object (mostly usage preferences like preferred/disliked actors or genres). We call hard constraints of the first type adaptation-sensitive hard constraints, whereas we refer to the second type as strict hard constraints.
We realized the actual expression of explicit hard constraints by preference tags containing an additional attribute *'constraint="hard"'*, which signalize the SIAM service (and all subsequent Web services) that this preference must not be violated.
Since in our framework all preferences are based on partial orders, standard skyline query evaluation is thus not

readily applicable. Though it is possible to render a total order from partial order preferences by considering the 'level' of objects in each preference graph (the level being defined by the longest path to any maximum value in the graph), by doing so we accept some inaccuracies induced by object incomparability.

Example (cont.): For instance, taking the actor preference of figure 3 we could transform the preference into a total order with 'Damon' as most important value (on level 1), followed by 'Willis' and 'Schwarzenegger' on the same importance level 2, and finally have 'Pitt' and 'diCaprio' as least preferred values again with same importance on level 3. But we would also state in the total order rendered from level information that 'Willis' is preferred over 'Pitt' though in fact they are incomparable in the partial order preference.

But from this total order induced by levels it is simple to derive a numerical assignment of score values for objects by using a utility function, translating the level information into simple scores. We normalize the score to a numerical value between 0 (least preferred) and 1 (most preferred). For each respective attribute value a in the database a score $S_A(a)$ with respect to attribute A in the query expressed by total order preference p can be computed, as follows:

$$S_A(a) := (max\_level + 1 - actual\_level)/(max\_level)$$

with max_level as the total number of levels in preference p and $actual\_level$ as the level on which attribute value a actually occurs with respect to A or $(max\_level + 1)$, if the value a is not explicitly stated in p.

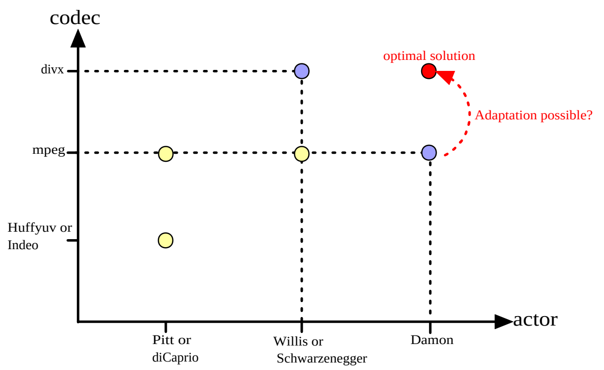

Fig. 6 Example Skyline for the Preferences of Figure 3

Example (cont.): Assume our user Kim requests a movie to be delivered to her PDA and states the preferences in figure 3. Further assume that the database contains only the following objects: a 'Matt Damon' movie in mpeg format (movie 1), an 'Arnold Schwarzenegger' movie in mpeg format (movie 2), a 'Bruce Willis' movie in divx format (movie 3), a 'Leonardo diCaprio' movie in Huffyuv format (movie 4) and a 'Brad Pitt' movie in mpeg format (movie 5). Figure 6 shows the resulting tuples. The skyline consists of only two objects (dark shaded): movie 1 and movie 3. An adaptation decision engine never needs to consider any other object from the database, i.e. movies 2, 4, and 5 (light shaded), because those are definitely dominated by the two skyline objects. For example, a 'Brad Pitt' movie will never be needed (not even as a base for subsequent adaptation), because it will always be dominated by the existing 'Bruce Willis' movie in divx format with respect to both dimensions.

However, please note that according to the user preferences the most preferred object would be a 'Matt Damon' movie in divx format (level 0 of the Pareto graph). But there simply is no instance in our database that satisfies this preference. In classical information systems the user query thus definitely has to be relaxed to the existing skyline objects. On the other hand, in multimedia systems it is often possible to derive an optimal object by suitable means. Though it is for instance not possible in our example to create the most preferred object by adapting the Willis/divx, movie because we obviously cannot replace the actors, it is indeed possible to transcode the Damon/mpeg movie into a Damon/divx movie (dotted arrow in figure 6), if there is a suitable transcoding service available that meets all hard constraints checked by the instantiation engine (e.g., QoS constraints). Let us summarize some different possibilities:

- Retrieve the Matt Damon movie in mpeg format and transcode it into divx format before delivery.
- Deliver the Bruce Willis movie in divx format.
- Deliver the Matt Damon movie in mpeg format.
- Retrieve the Arnold Schwarzenegger movie in mpeg format and transcode it into divx format before delivery.
- Deliver the Arnold Schwarzenegger movie in mpeg format
- ...

Now we will take a closer look of how to compute a ranking for the best adaptation decision based on the movies available in the database. Of course this ranking will also have to be according to the levels of the Pareto graph. Our scheme described in [3] presents a fair relaxation framework that can handle such problems. The framework describes how to relax soft constraints given by the user if no exact match should be available. First the decision engine has to build a hierarchy (representing the relaxation path) out of all given preference graphs like shown in figure 7.

The available tuples in our database are instances of the corresponding nodes (movie 1, movie 2, etc.). The best matching movie fulfills all soft constraints with a minimum amount of relaxation regarding a suitable quality measure. Since the different nodes in the preference graphs are already ordered according to the necessary

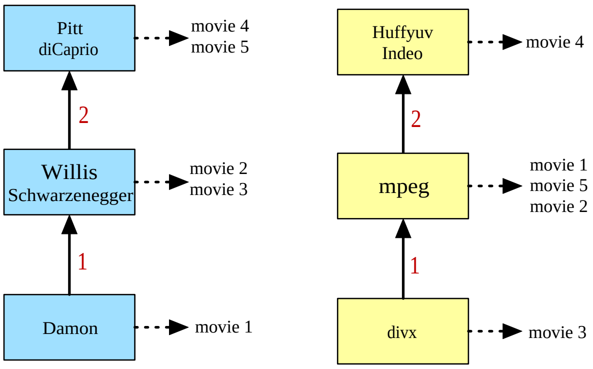

Fig. 7 Preferences Relaxation Paths

relaxation steps from the top or the bottom of the hierarchy or simply the relative distance to the root nodes. We can use a simple weighting factor to all edges in our hierarchy graphs to decide which movie to deliver. Our basic assumption for the computation of the weighting factor is, the close we get to the root of a relaxation path (i.e. the more relaxation is needed), the more unsatisfied the user will be with the delivered result. We thus need to derive a quality measure to compare between media objects.

Computation of the weighting factor (cf. [3]): Assume n preference graphs with their respective relaxation hierarchies. Every edge is labeled by $n^{(d-1)}$, where n is the number of preference graphs and d is the relative distance to the root node to achieve a fair, breadth first relaxation scheme. Now each database object o is labeled with weighting factor $wf(o)$ by summing up the labels of edges relaxed to find this object in each hierarchy. Sorting the object labels in ascending order leads to a preference order of media delivery decisions.

Example (cont.): In our example movie 1 and movie 2 both got the same weighting of 1 (cf. figure 7). If the objects have the same weightings and still no decision is possible, we have a random choice between them. Thus, aiming at a minimum necessary amount of relaxation using only the existing media according to we are left with two choices: delivering movie 1 (a Damon/mpeg movie) needs to relax the constraint on the codec by one level and delivering movie 3 (a Willis/divx movie) needs to relax the constraint on the actor by one level.
Before creating the final delivery order we have to take a look if it is possible to adapt one of the skyline movies to become a better weighting factor, respectively a more optimal solution. The potential adaptability is a-priory known for each attribute. We know that it is impossible to adapt the actor preference (figure 7, left) but it is possible to change the codec preference (figure 7, right) by a suitable transcoding service. Please note that the decision engine does not need to know, if such a service is actually available, because it only builds a hierarchy of the best possible adaptation order. Coming back to our example, we see that it is not possible to improve the weighting factor for movie 3, but it is possible to adapt movie 1 by transcoding. With a suitable transcoding service the weighting factor of movie 1* (Damon/divx) will be 0 and becomes indeed an instance of the optimal solution of our preference problem (cf. figure 6).

Please note that as soon as there are media objects already suitable for immediate delivery in the list, no more less important objects (i.e. not skyline objects) that have to be created by adaptation are needed. Hence not all database objects have to be ranked, but really only skyline objects or objects created by suitable adaptation from skyline objects. The space of possible adaptations is therefore limited. The following algorithm gives an overview about the whole procedure:

1. */* Extract preferences according to their attributes */*
   Given is the set P of preferences on n attributes in XML format by the user interaction tools.
   1.1. Extract the hard constraints for each attribute $HC_1, ..., HC_n$
   1.2. Divide $HC_1, ..., HC_n$ in adaptation-sensitive hard constraints $HC_1, ..., HC_j$ and the strict hard constraints $HC_{j+1}, ..., HC_n$
   1.3. Extract the soft constraints for each attribute $SC_1, ..., SC_n$
2. */* Transform partial orders $SC_i (1 \leq i \leq n)$ into total orders according to the respective level information. This step results in the preference relaxation graphs */*
   For each category $i (1 \leq i \leq n)$ do
   2.1. Compute max_level for SC_i as length of longest path
   2.2. For each attribute value a in SC_i do
      2.2.1. Compute $actual\_level$ for a as the maximum path length to a from some top object in SC_i
      2.2.2. Compute $S_A(a) :=$ $(max\_level + 1 - actual\_level)/(max\_level)$
   2.3. Build total order SC'_i using the ranking information given by $S_A(a)$
3. */* Determine possible media object set by skyline computation and compute respective relaxation weights */*
   Get initial set O of available media objects from the database using a skyline algorithm (e.g. from [4]) with SC'_i $(1 \leq i \leq n)$ as input preferences and filtered by strict hard constraints $HC_{j+1}, ..., HC_n$
   3.1. If $O = \emptyset$ then terminate with error
   / No suitable (or at least adaptable) media object is available in database */*
   3.2. For each object $o \in O$ do
      3.2.1. Compute weighting factor $wf(o)$ according to preference relaxation graphs SC'_i $(1 \leq i \leq n)$
   3.3. Set decision := false
   / no adaptation decision taken yet */*

4. */* Derive best adaptation decision */*
   While (not decision) do
   - 4.1. If $O = \emptyset$ then terminate with error
     / No media object adaptation meets hard constraints */*
   - 4.2. Choose set $D \subseteq O$ of objects with currently lowest $wf(o)$ $(o \in D)$ as best available objects
   - 4.3. If $wf(o) = 0$ $(\forall o \in D)$ then check adaptation-sensitive hard constraints $HC_1, ..., HC_j$
     / There is at least one perfect solution */*
     - 4.3.1. If $HC_1, ..., HC_j$ are satisfied for some $o \in D$ then set decision := true and return any such object o as adaptation decision
       / successful termination */*
     - 4.3.2. Else discard set D from O and if O becomes empty terminate with error
       / No media object is adaptable to meet hard constraints */*
   - 4.4. */* try to adapt all objects with lowest wf(o) to obtain some object o' with an even lower wf(o') */*
     For each $o \in D$ with $wf(o) > 0$ $(o \in D)$ do
     - 4.4.1. If o satisfies $HC_1, ..., HC_j$ then put o into set D' of possible adaptation decisions
     - 4.4.2. Try all possible adaptations of o as given by SIAM workflows [1] and $SC'_1, , SC'_n$ to get object o' with $wf(o') < wf(o)$
       - 4.4.2.1. If $HC_1, ..., HC_j$ are satisfied for o' then put o' into set D' of possible adaptation decisions, else discard o'
     - 4.4.3. Discard o from D and O
   - 4.5. If $D' \neq \emptyset$ get object x from D' with minimum $wf(x)$, set decision := true and return x as best matching adaptation decision
     / successful termination */*

The decision engine can now derive the final adaptation order by sorting the skyline objects and their possible adaptations as output by the algorithm according to their respective weighting factor. It then hands the respective list over to the service instantiation engine. This engine takes the sole responsibility for actually selecting the service chain by iterating over the adaptation list and checking individually whether a suitable chain can be instantiated at the current time.
However, one problem that may occur when using skyline query processing is that Pareto sets are known to grow exponentially in size with increasing number of independent preferences. As a remedy in [2] we introduced an approach to significantly reduce the skyline size by using so-called prime cuts. Prime cuts are interesting subsets of the full Pareto skyline, which are representative of the respective skyline and generally provide good compromises. The key to improved performance and reduced result sets sizes here is the relaxation of Pareto semantics to the concept of weak Pareto dominance (see [2] for details). The set of all non-weakly-dominated objects is referred to as the 'restricted' skyline.

In comparison to the full Pareto skylines our evaluation over a database with 25000 items in [2] shows, that by using restricted skylines the number of tuples only slightly increases, if more preferences are stated (see figure 8). Moreover, even for large numbers of preferences the restricted skyline can be computed about two orders of magnitude faster than traditional Pareto skylines (see figure 9). We can state that the restricted skyline approach makes even high-dimensional preference trade-offs (as may sometimes be necessary in personalized media adaptation) practical.

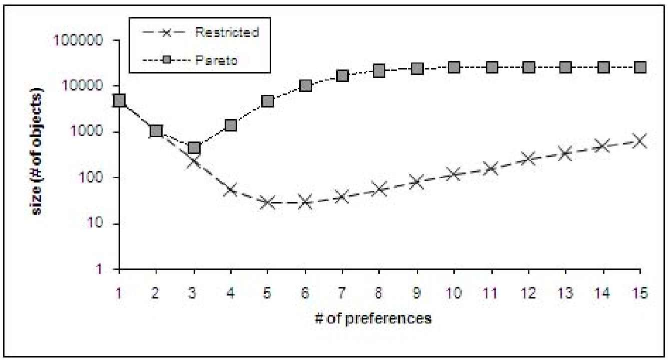

Fig. 8 Preference dimensionality and skyline size

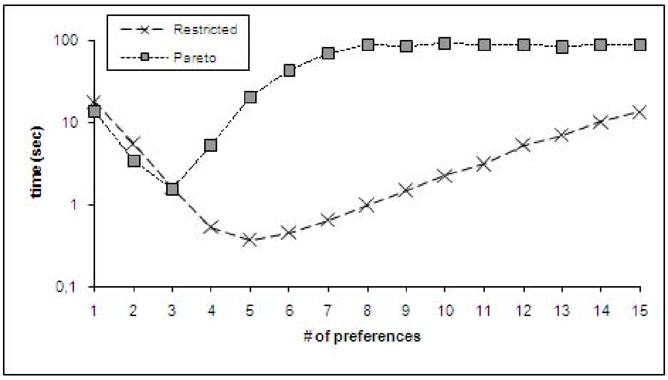

Fig. 9 Preference dimensionality and runtimes

6 A Media Streaming Test Case

As a proof of concept we implemented the key features of the decision engine in a small testbed (called PUMA) based on Web services. Since we were interested in deriving suitable adaptations our implementation focused specifically on the changes in the metadata documents and what could be derived as final adaptation decisions. We did not perform the actual adaptation of the video files in the testbed. All Web services have been developed

in Microsoft Visual Studio .NET in C#. For user interaction the SIAM (service instantiation and monitoring) service is addressed by a simple PHP interface, where users can place their initial requests with adequate preferences. Like discussed in section 4, the SIAM service runs the E^2Mon algorithm from [1].
Figure 10 shows the elements of our basic testbed. The experimental scenario focused different typical scenarios like the retrieval of videos from alternative databases, the merging with subtitles in different languages, as well as transcoding and quality adaptation. The initial query parameters, the user preferences and terminal capabilities are written into a MPEG-21 document, which is sent to the central SIAM service for further processing.

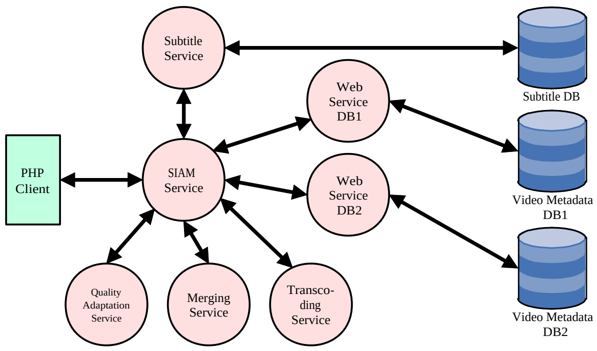

Fig. 10 Experimental Scenario

Figure 11 shows a screenshot of the PUMA testbed. Attribute values for (hard) content preferences or preference information in terms of profiles (like the PDA technical profile) can be submitted to the system. Objects are selected from the database accordingly and then prepared for further processing with an appropriate metadata description file stating all necessary details.

Example (cont.): Let us come back to user Kim who wants a movie streamed to her PDA with her preferences as discussed before and additionally requests a certain scaling according to her client device's display size specified in the PDA profile.

As discussed in section 5 the service instantiation engine gets the list from the adaptation decision engine and will choose an adequate workflow from a set of patterns. It first tries the transcoding of the 'Matt Damon' movie into divx format to deliver the best possible solution according to the preferences stated by Kim. To do this it has to integrate a suitable transcoding service into the delivery workflow. Since in most multimedia applications workflows can be anticipated and are well understood this task is performed by checking if an adequate transcoding service is available by requesting the functionality using a UDDI service directory. If the engine finds a suitable service, it will try to instantiate the corresponding service chain.

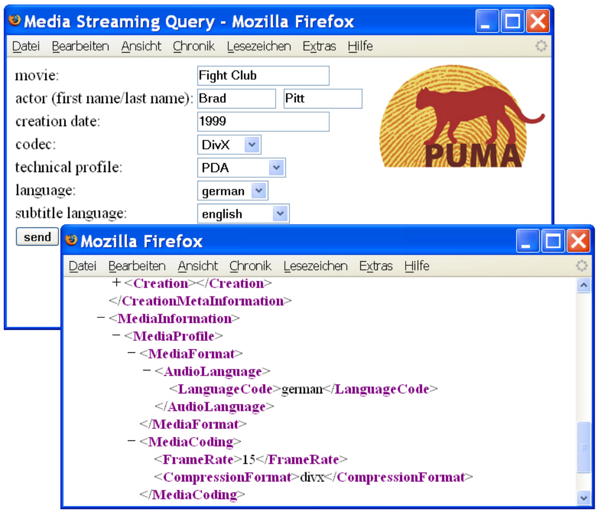

Fig. 11 Screenshot of the PUMA testbed

By handing over the preference constraints (including especially all hard constraints) the transcoding service is now enabled to check, if it is possible to transcode the movie into divx, e.g. matching statistics on the expected time needed for transcoding, etc. against the user-provided constraints. The service monitoring engine monitors the whole adaptation process. Hence, if the transcoding service decides that a transcoding is not possible or the transcoding fails, e.g. because the service is not longer available, a respective alarm is raised to the monitoring engine as described in detail in [1]. Now the service instantiation engine will try to find an equivalent service to recover from failure or decide that an adaptation of the Damon/mpeg movie to a Damon/divx movie is generally not possible. In this case it will try to instantiate the next alternative from the adaptation list.
If the transcoding succeeds, a scaling service (figure 9: quality adaptation service) is still needed, to adapt the current resolution of the movie to the highest possible display resolution of the PDA (see figure 2). This service again gets all preference data and decides locally if it is sensible to scale down the resolution. To make a decision about that, the Web service for example compares the information from the transcoding hints to the terminal capabilities. If a mismatch occurs, again an alarm has to be raised. Please note, that though some checks can already be performed by the adaptation decision engine, the services always may have to integrate local parameters into the decision whether to perform a task and what is the best way.
Moreover, in a typical workflow Kim may also request subtitles for a certain movie. The subtitles then can also be saved into the MPEG-21 metadata document and handed on to a merging service together with the original media from the database. After the merging process the service removes the now obsolete subtitles and the

language preferences from the metadata document of the adapted movie. and finally the adapted movie is delivered to user Kim.

Throughout our experiments it proved possible to handle the preference matchmaking and thus create a dynamic service composition based on the metadata with the SIAM service as central coordinator.

7 Conclusions and Future Work

With the increasing need of service oriented concepts in complex multimedia applications also the adaptation of content is split into several tasks. Depending on the available media, the discovered services and all semantic, user-specific and technical constraints, the selection of the best possible workflow needs to solve a complex preference problem. The current preference model in the MPEG-7/21 standard, however, is limited to the simple matching of numerical values representing the importance of each constraint. Hence for complex instantiations of workflows in a service-oriented adaptation engine a more sophisticated preference model is needed.

By integrating a model of partial order preferences and evaluating complex preference constraints using the skyline queries as currently explored by the database community, our framework is able to handle trade-offs in a more meaningful way. We have shown on the example of a typical media streaming scenario that an automatic adaptation of the multimedia content is possible by means of structured metadata information and how to efficiently explore the complex result meshes that emerge from combining independent content and adaptation preferences. These preferences are collected from independent sources such as explicit user preferences, intrinsic domain preferences, transcoding hints, or technical device capabilities. This complete preference information can then be augmented by all Web services in the workflow and thus allows an intelligent and highly personalized adaptation process. Moreover, every single Web service is enabled to decide how to adapt multimedia material best.

After implementing the key components as a proof of concept we are currently implementing the entire framework including the actual adaptation of video streams to get further insights on the scalability of our approach. The MPEG-21 standard already provides some description schemes to contain network QoS parameters, which will have a strong influence on the scalability of service-oriented adaptation processes. We want to explore in how far these parameters can be measured or estimated accurately enough to guarantee stable service compositions (especially since some parameters like e.g. network latency, may change very quickly and lead to the consideration of alternative service chains).

Another open question is, if the routing path should also be determined by preferences. Generally speaking, routing media can be performed in a custom-made overlay allowing the distribution of services along the routing path. Here, the effective management of trade-offs for all services involved can lead to a better overall service for end users. Finally, we plan to integrate higher level semantic descriptions into our framework. Recently the World Wide Web Consortium (W3C) started a Semantic Web task force [26] to investigate multimedia annotations. The respective standardization activities are promising to overcome the problem of missing mapping functions for MPEG-7/21.

References

1. W.-T. Balke, J. Diederich: A Quality- and Cost-based Selection Model for Multimedia Service Composition in Mobile Environments. In IEEE Int. Conf. on Web Services (ICWS), Chicago, IL, USA, 2006.
2. W.-T. Balke, U. Güntzer, W. Siberski: Getting Prime Cuts from Skylines over Partially Ordered Domains. In Database systems in Business, Technologie and Web (BTW 2007), Aachen, Germany, 2007.
3. W.-T. Balke, M.Wagner: Through Different Eyes - Assessing Multiple Conceptual Views for Querying Web Services. In World Wide Web Conference (WWW), New York, USA, 2004.
4. S. Börzsönyi, D. Kossmann, K. Stocker: The Skyline Operator. In International Conference on Data Engineering (ICDE), Heidelberg, Germany, 2001.
5. P Brusilovsky, A Kobsa, J Vassileva: Adaptive HyperText and Hypermedia. Kluwer Academic Publishers, USA, 1998.
6. J. Chomicki: Preference Formulas in Relational Queries. In ACM Transactions on Database Systems (TODS), Vol. 28 (4), 2003.
7. X. Gu, K. Nahrstedt, R. Chang, C. Ward: QoS-assured Service Composition in Managed Service Overlay Networks, In IEEE ICDCS 2003, Providence, RI, USA, 2003.
8. W. Kellerer, M. Wagner, W.-T. Balke, H. Schulzrinne: Preference-based Session Management for IP-based Mobile Multimedia Signaling. In European Transac-tions on Telecommunications, Vol. 15 (4), Wiley, 2004.
9. W. Kießling: Foundations of Preferences in Database Systems. In Int. Conf. on Very Large Databases (VLDB), Hong Kong, China, 2002.
10. A. Kobsa, W. Wahlster: User models in dialog systems. Springer, USA, 1990.
11. B. Köhncke, W.-T. Balke: Personalized Digital Item Adaptation in Service-Oriented Environments. In Int. Workshop on Semantic Media Adaptation and Personalization (SMAP), Athens, Greece, 2006.
12. H.-K. Lee, J. Nam, B. Bae, M. Kim, K. Kang, J. Kim: Personalized Contents Guide and Browsing based on User Preference. In AH2002 Workshop on Personalization in Future TV, Malaga, Spain, 2002.
13. C.-S. Li, R. Mohan, J. R. Smith: Multimedia Content Description in the InfoPyramid. In IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Seattle, WA, USA, 1998.
14. D. Mukherjee, E. Delfosse, J.-G. Kim, Y. Wong: Optimal Adaptation Decision-Taking for Terminal and Network Quality-of-Service. In IEEE Transactions on Multimedia, Vol. 7 (3), 2005.
15. P. Mylonas, M. Wallace: Using Ontologies and Fuzzy Relations in Multimedia Personalization. In Int. Workshop on Semantic Media Adaptation and Personalization (SMAP), Athens, Greece, 2006.

16. K. Nahrstedt, D. Xu: Finding Service Paths in a Media Service Proxy Network. In SPIE/ACM Multimedia Computing and Networking Conference (MMCN), San Jose, CA, USA, 2002.
17. K. Nahrstedt, W.-T. Balke: Towards Building Large Scale Multimedia Systems and Applications: Chal-lenges and Status. In ACM Workshop on Multimedia Service Composition (MSC), Singapore, 2005.
18. D. Papadias, Y. Tao, G. Fu, B. Seeger: An optimal and progressive algorithm for skyline queries. In ACM SIGMOD Conference (SIGMOD 03), San Diego, USA, 2003.
19. G. Stamou, J. van Ossenbruggen, J. Z. Pan, G. Schreiber: Multimedia Annotations on the Semantic Web. In IEEE Multimedia, Vol. 13(1), 2006.
20. B. L. Tseng, C.-Y. Lin, J. R. Smith: Using MPEG-7 and MPEG-21 for Personalizing Video. In IEEE Multimedia, Vol. 11(1), 2004.
21. C. Tsinaraki, S. Christodoulakis: Semantic User Preference Descriptions in MPEG-7/21. In Hellenistic Data Management Symposium (HDMS), Athens, Greece, 2005.
22. C. Tsinaraki, S. Christodoulakis: A User Preference Model and an Query Language that Allow Semantic Retrieval and Filtering of Multimedia Content. In Int. Workshop on Semantic Media Adaptation and Personalization (SMAP), Athens, Greece, 2006.
23. P. van Beek, A. B. Benitez, J. Heuer, J. Martinez, P. Salembier, J. Smith, T. Walker: MPEG-7 Multimedia Description Schemes (Version 3.1, Working Draft), ISO/IEC, 2000.
24. A. Vetro: MPEG-21 Digital Item Adaptation: Enabling Universal Multimedia Access. In IEEE Multimedia, Vol. 11 (1), 2004.
25. A. Vetro, C. Timmerer: Digital Item Adaptation: Overview of Standardization and Research Activities. In IEEE Transactions on Multimedia, Vol. 7 (3), 2005.
26. Word Wide Web Consortium (W3C): Multimedia An-notation on the Semantic Web.http://www.w3.org/2001/sw/BestPractices/MM/
27. M. Zhou, V. Aggarwal: An Optimization-based Approach to Dynamic Data Content Selection in Intelligent Multimedia Interfaces. In ACM Symposium on User Interface Software and Technology, Santa Fe, NM, USA, 2004.

Benjamin Köhncke currently is a Master student at University of Hannover and has received his BA in Computer Science at the University of Hannover, Germany in 2006.

Wolf-Tilo Balke since 2004 associate research director of L3S Research Center of University of Hannover. Before that he was a research fellow at the University of California at Berkeley, USA. He is a member of the Emmy-Noether-Program of the German Research Foundation. He has received his MS in mathematics and a PhD in computer science from University of Augsburg, Germany.