Efficient MPEG-21-based Adaptation Decision-Taking for Scalable Multimedia Content

Ingo Kofler[1a], Christian Timmerer[a], Hermann Hellwagner[a], Andreas Hutter[b], and Francesc Sanahuja[b]
[a] Klagenfurt University, Department of Information Technology, A-9020 Klagenfurt, Austria
[b] Siemens AG, Corporate Technology IC 2, D-81730 Munich, Germany

ABSTRACT

The MPEG-21 standard defines a framework for the interoperable delivery and consumption of multimedia content. Within this framework the adaptation of content plays a vital role in order to support a variety of terminals and to overcome the limitations of the heterogeneous access networks. In most cases the multimedia content can be adapted by applying different adaptation operations that result in certain characteristics of the content. Therefore, an instance within the framework has to decide which adaptation operations have to be performed to achieve a satisfactory result. This process is known as adaptation decision-taking and makes extensive use of metadata describing the possible adaptation operations, the usage environment of the consumer, and constraints concerning the adaptation. Based on this metadata a mathematical optimization problem can be formulated and its solution yields the optimal parameters for the adaptation operations. However, the metadata is represented in XML resulting in a verbose and inefficient encoding. In this paper, an architecture for an Adaptation Decision-Taking Engine (ADTE) is introduced. The ADTE operates both on XML metadata and on metadata encoded with MPEG's Binary Format for Metadata (BiM) enabling an efficient metadata processing by separating the problem extraction from the actual optimization step. Furthermore, several optimization algorithms which are suitable for scalable multimedia formats are reviewed and extended where it was appropriate.

Keywords: Adaptation Decision-Taking, MPEG-21, Digital Item Adaptation, Binary Format for Metadata (BiM)

1. INTRODUCTION AND MOTIVATION

The desire to gain access to advanced multimedia content anywhere/anytime and with any kind of device is nowadays growing tremendously. The research issues resulting from this development are generally referred to as Universal Multimedia Access (UMA)[1] and are well addressed within the research[2-6] as well as the standardization community[7,8]. In particular, the *device and coding format independent adaptation paradigm*[15] provides a flexible and generic methodology for adapting (scalable) multimedia content according to the usage environment. *Device independence* is guaranteed through a unified description of the environment in which the content is ultimately consumed or through which it is accessed. *Coding format independence* is accomplished by separating the adaptation decision-taking from the actual multimedia content adaptation. The former step comprises the selection of optimal parameter settings for multimedia content adaptation engines that satisfy constraints imposed by terminals and/or networks while maximizing Quality of Service (QoS). The latter step utilizes XML-based metadata describing the high-level syntax of a bitstream and adaptation thereof in the XML domain by applying remove and minor update operations. In a nutshell, XML is used for describing the syntax of a bitstream. The resulting document, called Bitstream Syntax Description (BSD), is then transformed, e.g., with an Extensible Stylesheet Language Transformations (XSLT) style sheet, and used by a generic processor to generate the adapted bitstream.

This paper addresses details of the adaptation decision-taking component which has been discussed very rarely in the literature. In particular, we focus on the adaptation decision-taking that is implied by the MPEG-21 Digital Item Adatpation (DIA) standard. MPEG-21 DIA uses an optimization-based approach which applies mathematical models for finding the maxima or minima of functions, possibly subject to constraints. To our knowledge, there exist two other approaches for adaptation decision-taking. The first approach comprises a simple look-up table mechanism[11] which

[1] ingo.kofler@itec.uni-klu.ac.at; phone +43 (463) 2700 3600; fax +43 (463) 2700 3699; www.itec.uni-klu.ac.at
Acknowledgments: Part of this work is supported by the European Commission in the context of the DANAE project (IST-1-507113). Further information is available at http://danae.rd.francetelecom.com.

holds possible adaptation parameters within a table representation and makes adaptation decisions using a table look-up. This approach is very simple and fast but lacks from flexibility. The second approach uses techniques from the artificial intelligence domain which views the problem of finding an optimal adaptation decision as a planning problem[12]. Nonetheless, this approach is rather new compared to the other approaches and a demonstration of its usage within the delivery chain is still outstanding.

The remainder of this paper is organized as follows. Section 2 briefly discusses adaptation decision-taking as standardized in the MPEG-21 multimedia framework. In Section 3 we introduce our architecture for a generic adaptation decision-taking engine (ADTE). The ADTE-related metadata handling, including results on compression efficiency and processing performance, is addressed in Section 4. The actual optimization process, implementation approaches, and associated evaluation results are discussed in detail in Section 5. Section 6 concludes the paper.

2. MPEG-21 DIGITAL ITEM ADAPTATION

The aim of the MPEG-21 standard, the so-called multimedia framework, is to enable transparent and augmented use of multimedia resources across a wide range of networks, devices, user preferences, and communities, notably for trading (of bits). As such, it introduces the concept of a Digital Item (DI) which is a structured digital object with a standard representation and metadata. Digital Items build the fundamental unit of transaction and distribution within the MPEG-21 framework. A vital and comprehensive part within MPEG-21 and with regard to UMA is part 7 of the standard, referred to as Digital Item Adaptation (DIA), which specifies normative description tools[2] to assist the adaptation of Digital Items. In particular, the DIA standard specifies means for enabling the construction of device and coding format independent adaptation engines. Note that only tools used to guide the engine are specified by DIA, the adaptation engines themselves are left open to industry competition. This paper describes an architecture for efficient adaptation decision-taking based on tools standardized within MPEG-21 DIA.

The Adaptation QoS tool can be used to describe the possible adaptations and parameters of the content and introduces two concepts for this purpose. The first concept are IOPins that are used to represent properties, adaptation parameters or a resulting quality. IOPins can be seen as a variable that is identified by a unique name and has a discrete or continuous domain. In the context of a MPEG-4 Bit-Sliced Arithmetic Coding (BSAC) audio stream possible IOPins can be the number of removed enhancement layers, the resulting bitrate and the quality of the adapted stream using an objective quality measure. The second concept that addresses the interrelationship between IOPins is called module and can be interpreted as a mathematical function. Within the DIA standard three different types of modules are specified: look-up tables, utility functions and stack functions. While the first and the second one are used to define functional dependencies by explicitly listing the function values for discrete function arguments, the third one allows the formulation of algebraic expressions in postfix notation. As an example, a look-up table could be used to describe the functional dependency between the number of enhancement layers removed and the resulting bitrate. IOPins can be distinguished into two disjoint sets, based on their usage within the modules. While the values of dependent IOPins are determined by a functional dependency, the values of the free IOPins can be chosen arbitrarily. For the BSAC example this means that the IOPin representing the number of enhancement layers to remove is a free one, while the resulting bitrate is a dependent IOPin.

The tool allowing for device independence is generally referred to as Usage Environment Description (UED). The UED provides a fundamental input to any adaptation engine and includes means for describing terminal capabilities and network characteristics as well as user characteristics and the characteristics of the natural environment. To enable users and providers to further constrain the usage of a Digital Item, the Universal Constraints Description (UCD) tool has been specified. With this tool, it is possible to describe two types of constraints that impact the adaptation process[14]. Limitation constraints can be used to constrain the adaptation decision, e.g., by preventing the resulting bitrate of a resource to be higher than the network's nominal bandwidth. The limitation constraints are formulated as boolean expressions that have to be satisfied for a valid adaptation decision. Additionally, several optimization criteria can be specified to guide the adaptation decision. In the MPEG-21 DIA terminology these optimization criteria are called optimization constraints and are also represented as expression in postfix notation (stack function). In order to formulate constraints it is also necessary to reference both values of the UED (e.g., available bandwidth, screen resolution) and the

[2] A tool within MPEG-21 DIA is defined as the XML-based syntax of the description formats and its corresponding semantics written in natural English.

IOPins within the Adaptation QoS description. This issue is implemented by references that can either point to a certain Universal Resource Identifier (URI) or reference values by their semantic through a Uniform Resource Name (URN).

The decision-taking can be performed by solving an optimization problem that can be derived from the metadata. The goal is to find an assignment of values for the IOPins that does not violate the given limitation constraints and is optimal concerning the specified optimization constraints. However, the finding of an optimal assignment of IOPins is limited to the free IOPins as the dependent IOPins are calculated based on the values assigned to the free ones. The generic design of AdaptationQoS and UCD spans a *mixed-variable multi-criteria optimization problem with general constraints*. The problem belongs to the class of mixed variable problems as IOPins can be either discrete or continuous. In practice, however, for scalable bitstreams continuous IOPins are used very rarely. For example, one would never truncate two and a half enhancement layers of a scalable bitstream. Nonetheless, one could argue that a continuous representation is necessary for fine-grain scalability but adaptation is still performed in discrete steps though.

3. ADTE ARCHITECTURE

This section introduces an architecture for an efficient MPEG-21-based adaptation decision-taking engine (ADTE). Its task is to process the DIA descriptions in order to find optimal parameters for the content adaptation. It should be kept in mind that MPEG-21 only standardizes the XML descriptions, and so the ADTE introduced here is not normative and is only one among many possible design and implementation approaches. Its architecture (Fig. 1) emerged from the idea to separate the metadata handling from the actual optimization process. A further intention during the design of the ADTE was the support of a variety of usage scenarios, i.e., the hybrid mode[15] turned out to be the desirable adaptation strategy.

The *problem extractor* is responsible for metadata handling. It takes the AdaptationQoS description and one or several UCDs as input, processes them and generates an internal mathematical representation, i.e., the *problem description*. This problem description is an object-oriented data structure and mainly consists of the IOPins, their interrelationships (i.e., modules), and limitation as well as optimization constraints. Both UCDs and AdaptationQoS descriptions can contain references to IOPins or values taken from the usage environment description (e.g., a reference to the available network bandwidth). The handling of these references is done by the *reference manager* which tightly works together with the problem extractor. Each time the problem extractor encounters a reference it invokes the reference manager that creates an internal reference representation. These internal representations can be seen as initial empty placeholders, which are filled in a subsequent step. This filling is performed by the *reference resolver* under control of the reference manager. In particular, the reference manager maintains a set of XPath expressions according to the semantics defined in MPEG-21 DIA. The reference resolver processes the UED and determines the referenced values by evaluating the corresponding XPath expression. Furthermore, it detects invalid references, i.e., references to values in the usage environment description that are not available, and modifies the optimization problem by removing the affected constraints. The output of this component is therefore denoted as the *modified problem description*. Finally, the *optimizer* is responsible for finding a solution for the modified problem description by applying a mathematical algorithm. As within MPEG-21 no optimization method is explicitly specified, each algorithm that is suitable for a mixed-integer multi-criteria optimization problem can be used in this step. The result is an assignment of values for the IOPins that is feasible concerning the limitation constraints and optimal with respect to the optimization constraints. The output is delegated to an adaptation engine which uses the values of the IOPins as parameters for the adaptation.

The main benefit of this architecture is the separation of the problem extraction, the reference handling and the solving of the optimization problem. The usage of an internal problem representation enables the optimizer to fulfill its task independently of the encoding format of the metadata. That is, the initial metadata can be either encoded as plain text

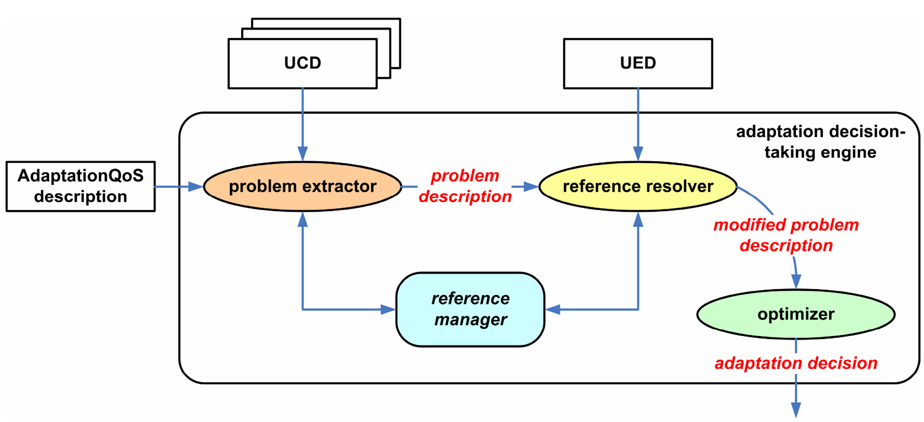

Fig. 1. MPEG-21-based Adaptation Decision-Taking Engine Architecture

XML or by applying a more efficient binary encoding scheme like BiM. Following this approach, the architecture supports usage scenarios where the decision-taking can be based on both metadata encoded in BiM and in plain text XML. It also supports the idea of the hybrid mode by supporting an arbitrary number of UCDs. The problem extractor processes them and generates a set of limitation and optimization constraints. Different priorities are assigned to optimization constraints, depending on the origin of the constraint. Although not foreseen in the standard, in our architecture the constraints that are formulated by the consumer have higher precedence than those issued by the content provider. The priorities can optionally be used by the subsequent optimizer.

In order to validate this approach, we implemented a prototype of an ADTE following this architecture. The implementation is based on C++ using Microsoft Visual Studio 6.0 as development environment. Different implementations of the problem extractor and the reference resolver based on XML and BiM parsers were developed for the metadata processing. Furthermore, a variety of optimization algorithms were implemented and evaluated concerning their suitability for the adaptation decision-taking. A detailed description of the different implementations and the resulting performance is given in the subsequent sections.

4. METADATA HANDLING

Within MPEG-21 metadata plays a vital role to achieve an interoperable multimedia framework. However, the usage of XML-based metadata comes along with some disadvantages when transmitting the metadata over the network. In particular XML metadata tends to be quite verbose resulting in significant metadata overhead. Additionally, XML metadata lacks of streaming capabilities that are required when streaming multimedia content. Previous research[10] showed that BiM is a suitable binary encoding scheme for XML metadata, that offers both streaming and fragmentation capabilities and achieves compression rates superior to traditional compression algorithms for MPEG-21 metadata. Therefore, we evaluated the usage of BiM to encode the metadata related to the adaptation decision-taking.

4.1 Use Cases

As a basis for the evaluation and the measurements we defined three different use cases in the context of scalable bitstreams. The first use case describes MPEG-4 visual elementary streams (i.e., *akiyo and foreman* sequences). The second use case deals with the adaptation of an MPEG-4 BSAC audio stream (*jm*) and the third use case represents the adaptation of JPEG2000 images (*city, shanghai*). Each use case comprises an AdaptationQoS description that describes the adaptation possibilities and the resulting utilities of the adapted multimedia content, and two UCDs that represent constraints of the consumer and the provider of the multimedia content, respectively. All descriptions are initially stored in separate XML files, which are valid against the schema specified within MPEG-21 DIA. In this paper we use the following naming conventions for the metadata, i.e., the name of the actual sequence followed by an underscore and a suffix, which is either *aqos* for the AdaptationQoS description, *ucd_provider* for the UCD containing the provider's constraints or *ucd_consumer* for the UCDs of the consumer. In addition to the content-related metadata different UEDs covering the characteristics of three types of terminals (i.e., personal computer, personal digital assistant, and mobile phone) were created. Depending on the type of the terminal they are describing, the UEDs are named *ued_pc*, *ued_pda* and *ued_mobile*.

4.2 Compression efficiency

The first part of the evaluation focuses on the achievable compression ratio. Therefore, the descriptions that are initially encoded in plain text XML are compressed using BiM. Each description is encoded as one access unit (AU) consisting of exactly one fragment update unit (FUU)[9]. As during the problem extraction the whole description has to be examined, a fragmentation into more than one FUU was not reasonable for our use cases and would only result in an encoding overhead. As encoding strategy for strings, the string tokenization codec[18] is utilized. Note that in streaming scenarios AdaptationQoS information may be provided per bitstream segment, i.e., one AU/FUU per bitstream segment.

The results for the AdaptationQoS descriptions can be found in Table 1. The comparison shows that the encoding with BiM leads to a significant reduction in file size. However, the compression factor that is achieved is highly dependent on the structure of the AdaptationQoS description. The lowest reduction in size can be observed for the *city_aqos* and *shanghai_aqos* descriptions. This comparatively low compression factor of roughly about 5 can be explained by the use of a look-up table and a utility function, that describe 36 and 60 different adaptation operations by listing them explicitly. BiM encodes these parts of the description as list of numerical values without further compression which negatively influences the compression ratio. In contrast to that, higher compression factors of about 8 are achieved for the *foreman_aqos* and *akiyo_aqos* descriptions that do not extensively use look-up tables. The figures show that BiM

Table 1. AdaptationQoS descriptions

Description	Size XML [Bytes]	Size BiM [Bytes]	Compression factor
city_aqos	2986	559	5.342
shanghai_aqos	2892	529	5.467
jm_aqos	1404	259	5.421
foreman_aqos	2518	308	8.175
akiyo_aqos	2534	309	8.201

Table 2. Usage Environment Descriptions (UED)

Description	Size XML [Bytes]	Size BiM [Bytes]	Compression factor
ued_pc	4357	310	14.055
ued_pda	4124	287	14.369
ued_mobile	3221	233	13.824

Table 3. Universal Constraint Descriptions (UCD)

Description	Size XML [Bytes]	Size BiM [Bytes]	Compression factor
city_ucd_consumer	2822	243	11.613
city_ucd_provider	1607	192	8.370
shanghai_ucd_consumer	2822	243	11.613
shanghai_ucd_provider	1619	195	8.303
jm_ucd_consumer	1811	198	9.146
jm_ucd_provider	728	112	6.500
foreman_ucd_consumer	1579	186	8.489
foreman_ucd_provider	630	84	7.500
akiyo_ucd_consumer	1571	186	8.446
akiyo_ucd_provider	834	134	6.224

performs better in the case of UCD descriptions (Table 3). The majority of the descriptions can be compressed with a compression factor between approximately 8 and 12. This can be explained by the structure of the UCDs that only contain stack functions as constraints, which benefit more from the schema awareness of BiM. However, the best results concerning the compression efficiency can be achieved when using BiM for encoding UED descriptions (Table 2). The average compression factor is about 14, which is significantly higher than the AdaptationQoS and UCD results. This is based on the fact that the normative UED schema introduces quite verbose attribute and element names which can be efficiently encoded using BiM.

4.3 Processing performance

The figures show that BiM represents an effective encoding alternative for the descriptions related to the adaptation decision-taking. In general the advantage of having smaller file sizes by applying compression is accompanied with the disadvantage of processing-intensive compression and decompression algorithms. However, documents that are encoded using BiM can be processed in the binary domain without explicitly decompressing them in advance[18]. To investigate the impact of the BiM encoding on the processing performance, we developed three different implementations of the problem extractor component which are operating on plain text XML and compressed metadata, respectively.

We studied the processing time of a problem extractor component called *AQoSExtractor*, which is responsible for processing the Adaptation QoS metadata and generating the internal problem description. Two of the three different implementations under investigation are operating on XML metadata and are based on the Simple API for XML (SAX) and the Document Object Model (DOM) paradigms, respectively. Both the SAX and the DOM approaches utilize the open source parser Xerces v2.7. Third implementation operates on descriptions encoded using BiM and uses a BiM-specific software component called BAX[18] parser (BiM API for XML). It follows the push paradigm and makes use of a document handler in a way comparable to the SAX approach. However, the main differences are the handling of numerical values and the indication of the actual element. Instead of providing the values of the attribute or the element's content as strings, the BAX parser enables direct access to the binary representation of their values.

The three implementations of the AQoSExtractor are compared concerning the processing time that is required to parse the metadata and generate the internal problem representation. The AdaptationQoS descriptions are processed directly in main memory, i.e., I/O issues are deliberately neglected. Furthermore, each implementation was evaluated 100 times and the mean value of the measured processing time was taken as the final value of the measurement. All tests were performed on a laptop with an Intel Pentium 4 Processor running at 1.8 GHz and with 512 MB main memory. As operating system Microsoft Windows XP Service Pack 2 was used. The resulting average processing times of the three implementations are listed in Table 4. The figures show that the BAX based implementation, which is available only as prototype and is not optimized concerning performance, is only 50 to 75 percent slower than the SAX implementation. Compared to the DOM approach, the BiM implementation is slightly slower - about 10 percent - in the JPEG2000 and BSAC use cases. However, the BAX parser performs almost equally well in the video use cases, which is attributed to the different structure of the AdaptationQoS descriptions. Based on the competitive processing times and the excellent compression ratios we can infer that the usage of BiM leads to a more efficient adaptation decision-taking. Furthermore,

Table 4. Comparison of the processing times of the AQoSExtractor implementations

Description	SAX-based impl. [ms]	DOM-based impl. [ms]	BAX-based impl. [ms]
city_aqos	7.4	9.8	11.3
shanghai_aqos	7.3	9.5	11.1
jm_aqos	2.0	3.3	3.5
foreman_aqos	2.8	4.9	4.9
akiyo_aqos	2.9	5.1	5.1

BiM comes along with the streaming and fragmentation functionality, which can be exploited to efficiently signal changes of the usage environment by updating only the dynamic parts of the UED, e.g., the available network bandwidth.

5. OPTIMIZATION

The adaptation decision-taking within MPEG-21 is based on solving the optimization problem as introduced in Section 2. This mathematical approach leads to a generic processing model within the ADTE that is independent of the actual type of the Digital Item and purely relies on metadata. Following the architecture introduced in this paper, this task is up to the optimizer component that does not have to process the metadata and can focus on the optimization task itself.

Unfortunately the optimization problem yields some properties that make the problem solving difficult. The functional dependencies between the IOPins can be expressed by look-up tables and utility functions. As a consequence, the resulting function is not continuous and only defined at certain values. Therefore, analysis-based optimization algorithms that rely on derivatives of the function cannot be applied. Additionally, the limitation constraints that restrict the feasible solution space are expressed as stack functions, which evaluate to true or false. The DIA standard does not envision any restrictions on these stack functions, like allowing only relational operators. Instead, the limitation constraint can be arbitrarily complex, e.g., containing case differentiations and algebraic expressions. Therefore, the optimizer cannot make any assumptions about the limitation constraints and must handle both the limitation constraints and the modules as black-boxes. Further important parts of the optimization problem are the optimization constraints that are also expressed as stack functions without any restrictions of their shape. In the simplest form the constraint only contains a reference to an IOPin that has to be maximized or minimized. But also stack functions containing several IOPins that are combined in a mathematical formula are possible. The second issue in this context is the possibility of defining multiple optimization constraints that can even be contradictory. The class of optimization problems that can be formulated is known as mixed-integer multi-criteria optimization problem. Algorithms for solving this problem were already discussed in the current literature[13]. The most promising algorithm for solving the problem is the Mesh Adaptive Direct Search (MADS)[17], which is an iterative search algorithm. Unfortunately, not all of the features offered by MPEG-21 DIA can be exploited when using MADS, e.g., equality constraints are not supported. However, in the case of scalable bitstreams it is very unlikely to face non-discrete IOPins as the adaptation of the content can only be performed in discrete steps. As a result of our work in the context of scalable bitstreams, we focused on discrete optimization problems.

The simplest approach that can be applied to discrete problems is an exhaustive search within the problem space which is also known as *generate-and-test*. Therefore, all possible value assignments for free IOPins are generated and the dependent IOPins are calculated by using the functional dependencies defined by the modules. Afterwards the limitation constraints are tested. Each assignment that satisfies the limitation constraints is considered to be a solution candidate. The final solution is selected among these candidates depending on the number of optimizations constraints. In the case where no optimization constraints are defined, each of the candidates can be taken as a solution. A single optimization constraint leads to an evaluation of the stack function for each candidate. The candidate with the best function value is finally taken as the solution for the optimization problem. The selection of a solution is clearly more complicated in use cases where the optimization is guided by two or more constraints. Appropriate strategies for this scenario are discussed below. A drawback of this approach is the runtime complexity that depends on the number of free IOPins and the cardinality of their domains. However, the optimization problems derived from use cases in the context of scalable bitstreams have a limited number of IOPins (i.e., depending on the scalability dimensions of the bitstream), typically resulting in fewer than thousand different adaptation possibilities. One can assume that this amount of possibilities should be easily generated and evaluated on a modern computer even in time critical streaming scenarios. In order to speed up the optimization we further modified the initial algorithm by checking relevant limit constraints before calculating the dependent IOPins. Although this modification has no influence on the complexity class of the algorithm it reduces the number of module calculations and limitation constraint checking.

Within the *generate-and-test* approach different strategies can be applied to select an appropriate adaptation decision based on the candidate solutions. In the following we evaluated three possible strategies namely Pareto optimality, a weighted sum approach and a selection algorithm based on priorities.

The concept of Pareto optimality, which is suggested for handling multiple optimizations constraints[13], is based on dominating points. The proposed algorithm[13] incrementally constructs a set of Pareto optimal points where each of them can be used as a solution for the optimization problem. However, this approach does not work well when dealing with a problem that contains contradictory optimization constraints. The simultaneous maximization and minimization of an IOPin can lead to points that do not dominate each other, resulting in a long list of dominating points. As contradictory constraints cannot be eliminated, it does not seem to be an adequate strategy for the decision-taking. Another approach for handling multi-criteria optimization problems is to create one new objective function by combining the initial functions o_i in a weighted sum. The resulting function is also known as preference function[16] and should represent the trade-off between different objectives. The new objective function o' for an IOPin assignment v can be written as weighted sum $o'(v)=w_0o_0(v) + w_1o_1(v) + \ldots + w_{n-1}o_{n-1}(v)$. In order to combine the different optimization constraints in a fair way, the weightings have to consider the range of each function. It is recommended to choose a value for w_i so that the corresponding summand only takes values within the interval [0, 1]. Finding an appropriate weighting is easy for functions consisting of a single IOPin with a known domain. For handling more complex objective function an analysis of the function concerning its range is necessary to obtain a correct weighting, which is nearly as hard to achieve as the optimization problem itself. Therefore, this strategy is not advisable for each optimization problem. However, the advantage of this approach is the handling of conflicting optimization constraints because a minimization and a maximization constraint concerning the same IOPin are compensated.

A further approach for handling multiple optimization constraints is to introduce priorities and to sort out the feasible candidates in a hierarchical manner. As the DIA standard does not support priorities among the optimization constraints, a priority scheme is applied within the problem extractor component as already introduced in Section 3. Based on the priorities and the resulting order of the optimization constraints, a hierarchical selection algorithm can be applied to the solution candidates. The input to the selection algorithm is a set of solution candidates C and a vector of objective functions. The algorithm iterates through all objective functions, starting at the one with the highest priority. At each iteration i the objective function o_i is applied to each candidate in C. Based on the resulting function value the candidates are inserted into a temporary collection C_{temp} in ascending order. At the end of each iteration i the best b_i candidates are taken from C_{temp} and form the candidates C that are used in the subsequent iteration $i+1$. In order to guarantee a fair selection process the values of b_i should be descendent and b_n should be 1. The behavior of the algorithm concerning the degree of priority awareness can be controlled by the values of $b_1 \ldots b_n$. However, the values should always depend on the initial number of candidates. As the number of candidates has to be known in advance, it cannot be applied within the *generate-and-test* loop. In fact the only task of the candidate handling in the loop is to create the set of candidates, which is input to the selection process afterwards. Therefore, all candidates have to be kept in memory, which can be seen as disadvantage. However, as the approach can handle contradictory optimization constraints by privileging the consumer's constraints and does not rely on the shape of the optimization constraints like the weighted sum approach, it can be suggested as the best strategy for the adaptation decision-taking task at hand.

We implemented the three different strategies and measured the processing time of the optimizer for the three use cases using the same environment as introduced in Section 4. The tests were executed 100 times and the mean value of the measured times was taken as final value. As expected the tests showed that the processing time highly depends on the use case and on the number of possibilities to investigate. However, for the shanghai use case, which yields the highest number of 360 possibilities, the solution was found in less than 5 ms. Therefore, we infer that although the algorithm scales very bad for a high number of possibilities the *generate-and-test* approach is sufficient for typical adaptation decisions. Additionally, the comparison of the three strategies (Pareto optimality, weighted sum and priority-based) showed that there are no major differences in their runtime.

6. CONCLUSIONS

In this paper we propose an architecture for an MPEG-21-based adaptation decision-taking engine that separates the tasks of extracting an optimization problem from the metadata, resolving references to the usage environment description and solving the problem. This modular approach allows the usage of both XML and BiM metadata and supports the combination of different implementations to design an ADTE for special use cases. We showed that the overhead caused by XML metadata can be reduced significantly by using the Binary Format for Metadata (BiM). Compression factors between 5 and 14 can be achieved for typical DIA descriptions. The loss of performance when operating on the compressed data with the BAX parser is about 60 percent compared to a mature SAX parser. It should be kept in mind that the BAX parser is only a prototype and surely offers improvement capabilities to reduce this performance gap. Concerning the actual decision-taking we focused on discrete optimization problems as they are most common for scalable bitstreams. We demonstrated that an exhaustive search in the solution space (*generate-and-test*) is sufficient for typical optimization problems in the context of scalable bitstreams. Multiple optimization constrains should be handled by a priority-based approach where consumer constraints are handled with higher precedence.

REFERENCES

1. A. Vetro, C. Christopoulos, and T. Ebrahami, eds., *IEEE Signal Processing Magazine, special issue on Universal Multimedia Access*, vol. 20, no. 2, March 2003.
2. Y. Su, J. Xin, A. Vetro, and H. Sun, "Efficient MPEG-2 to H.264/AVC Intra Transcoding in Transform-domain", *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2005.
3. M. van der Schaar and Y. Andreopoulos, "Rate-Distortion-Complexity Modeling for Network and Receiver Aware Adaptation", *IEEE Transactions on Multimedia*, vol. 7, no. 3, June 2005, pp. 471-479.
4. J. Song, S. Yang, C. Kim, J. Nam, J.W. Hong and Y.M. Ro, "Digital Item Adaptation for Color Vision Variations", *Proceedings SPIE Conf, on Human Vision Electronic Imaging VIII*, Santa Clara, CA, January 2003.
5. B. Feiten, I. Wolf, E. Oh, J. Seo and H.-K. Kim, "Audio Adaptation According to Usage Environment and Perceptual Quality Metrics", *IEEE Transactions on Multimedia*, vol. 7, no. 3, June 2005, pp. 446-453.
6. D. Mukherjee, A. Said, and S. Liu, "A framework for fully format-independent adaptation of scalable bit-streams," *IEEE Transactions on Circuits and Systems for Video Technology*, Special Issue on Video Adaptation, vol. 15, no. 10, October 2005, pp. 1280-1290.
7. P. Salembier and J. R. Smith, "MPEG-7 multimedia description schemes", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, 2001, pp. 748-759.
8. A. Vetro, "MPEG-21 Digital Item Adaptation: Enabling Universal Multimedia Access", *IEEE Multimedia*, vol. 11, no. 1, January-March 2004, pp. 84-87.
9. O. Avaro and P. Salembier, "MPEG-7 Systems: overview", *IEEE Transactions on CSVT*, vol. 11, no. 6, 2001, pp. 760-764..
10. C. Timmerer, I. Kofler, J. Liegl, and H. Hellwagner, "An Evaluation of Existing Metadata Compression and Encoding Technologies for MPEG-21 applications", *Proceedings of the First IEEE International Workshop on Multimedia Information Processing and Retrieval (IEEE-MIPR 2005)*, Irvine, California, USA, December 2005, pp. 534-539.
11. L. Böszörményi, H. Hellwagner, H. Kosch, M. Libsie, S. Podlipnig, "Metadata Driven Adaptation in the ADMITS Project", EURASIP Signal Processing: Image Communication Journal, Vol. 18, No. 8, Sept. 2003 (Special Issue on Multimedia Adaptation), pp. 749-766.
12. D. Jannach, K. Leopold, C. Timmerer, and H. Hellwagner, "A Knowledge-based Framework for Multimedia Adaptation", *Applied Intelligence - Special Issue: Innovations in Applied Artificial Intelligence*, vol. 24, no. 2, pp. 109-125, 2006.
13. D. Mukherjee, E. Delfosse, J.-G. Kim, and Y. Wang, "Optimal Adaptation Decision-Taking for Terminal and Network Quality-of-Service", *IEEE Transactions on Multimedia*, vol. 7, no. 3, June 2005, pp. 454-462.
14. D. Mukherjee, G. Kuo, S. Liu, and G. Beretta, "Proposal for a Universal Constraints Descriptor in DIA", *ISO/IEC JTC1/SC29/WG11 M9772*, Trondheim, July 2003.
15. C. Timmerer, S. Devillers, and A. Vetro, "Digital Item Adaptation – Coding Format Independence", in: I. S. Burnett, F. Pereira, R. Van de Walle, and R. Koenen (eds.), The MPEG-21 Book, John Wiley and Sons, May 2006, pp. 282-331.
16. Y. Sawaragi, H. Nakayama, and T. Tanino, "Theory of multiobjective optimization", Academic Press Inc., 1985.
17. C. Audet and J.E. Dennis, "Mesh Adaptive Direct Search Algorithms for Constrained Optimization", Technical Report TR04-02, CAAM, Rice University, Houston, USA, January 2004.
18. C. Timmerer, T. Frank, and H. Hellwagner, "Efficient processing of MPEG-21 metadata in the binary domain", *Proceedings of SPIE International Symposium ITCom 2005 on Multimedia Systems and Applications VIII*, Boston, Massachusetts, USA, October 2005