O

OPTIMIZATION-BASED MULTIMEDIA ADAPTATION DECISION-TAKING

Ingo Kofler, Christian Timmerer, and Hermann Hellwagner
Department of Information Technology, Klagenfurt University, Austria

Synonym: *Finding parameters for multimedia content adaptation utilizing optimization techniques.*

Definition: *Optimization-based multimedia adaptation decision-taking is referred to as the process of finding the optimal selection of parameter settings for the actual multimedia content adaptation engines that satisfy given constraints while maximizing Quality of Service (QoS).*

Problem Description

Multimedia content adaptation is regarded as a key technology to enable the vision of Universal Multimedia Access (UMA) [1]. UMA refers to the idea that multimedia content can be consumed anytime and anywhere, regardless of the end device and the delivery networks. Since the adaptation of a multimedia content can be performed along different dimensions, e.g., a video can be reduced in either spatial or temporal resolution or a combination of both, a variety of adaptation possibilities arise. The task of adaptation decision-taking is to determine which adaptation operations should finally be performed on the multimedia content in order to close the mismatch between the initial content and the final content variation that suits, e.g., device and network capabilities while maximizing Quality of Service (QoS). The outcome of this decision-taking process can be seen as a set of adaptation parameters that steer the actual multimedia content adaptation engines.

The high-level architecture of a multimedia content delivery framework is depicted in **Figure 1**. A central module of this framework is the *adaptation decision-taking engine* (ADTE) which is the main topic of this article. Its task is to perform adaptation decision-taking as defined above, i.e., to provide the optimal parameter settings for the actual *multimedia content adaptation engine* that satisfy the given constraints while maximizing QoS. The input to the ADTE is a set of constraints which can be divided into two categories. The first category comprises constraints that emanate from the *content consumer* and are transmitted to the *multimedia service provider;* these are generally referred to as *usage environment description* or *context description*. The second category includes information pertaining to the actual *multimedia content* and is simply known as *multimedia content description*.

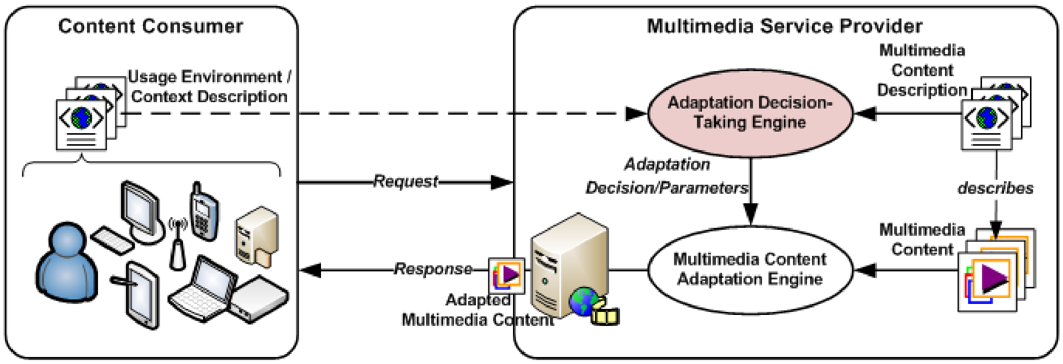

Figure 1. High-Level Architecture of a Multimedia Content Delivery Framework.

The multimedia content description may provide the relationships between the properties of the multimedia content, constraints, feasible adaptation operations/parameters satisfying these constraints, and associated utilities (qualities). Thus, the multimedia content description may ease the adaptation decision-taking as it will be described in the following sections.

Technical Solution Approaches

One approach for realizing adaptation decision-taking is to model the problem of finding adaptation parameters as a mathematical optimization problem [2, 3]. From a conceptual point of view this requires the following modeling steps:

1. Adaptation parameters and properties of the multimedia content are modeled as variables with a certain domain. The domain can be either discrete or continuous. While discrete domains contain only a finite set of values, continuous domains include all real values within a given interval.
   Examples: The parameter for the horizontal resolution of a video can be represented by a variable *HorizResolution* with a discrete domain containing the values 1920, 720 and 640. Similar to the adaptation parameters, relevant content properties are also represented by variables, e.g., the *BitRate* which could take values from the interval [0.5, 20] Mbps.

2. There may be dependencies among the parameters and properties represented by variables. This means that the selection of one or more parameters can have an influence on another variable, e.g., a certain value for the horizontal resolution of a video will have an impact on the bitrate of the video. Although some of these dependencies might be obvious, they have to be modeled explicitly for decision-taking. The functional dependency between variables can be either expressed by specifying all parameter-value tuples explicitly (e.g., as a kind of list or look-up table) or by defining an algebraic expression. The decision of how the functional

dependency is defined, depends both on the use case and on the domains of the variables involved.
Examples: The first way might be appropriate for defining the dependency between the horizontal resolution and the resulting bitrate, e.g., BitRate = f(HorizResolution) = {1920→15, 1024→5, 640→3.5}. On the other hand, an algebraic expression might be useful to define the relationship between the horizontal and vertical resolution, when considering that the aspect ratio should be preserved. In the case of a 16:9 aspect ratio the functional dependency might look as follows: VertResolution = f(HorizResolution) = HorizResolution/16*9.

3. Once the variables and their relationships are identified and modeled, constraints on the possible values (i.e., possible adaptation parameters) of the variables are defined. These constraints provide means for imposing restrictions on adaptation parameters and/or content properties with regard to limitations of the usage context (i.e., end user device, network, etc.). Constraints can be formulated as Boolean expressions that evaluate to true or false. In the context of decision-taking typically not a single constraint but a set of constraints is specified. The implication is that for a valid adaptation decision all constraints have to be satisfied, i.e., evaluate to true; the parameters are then referred to as feasible adaptation parameters.
Example: Let's assume we have an available bandwidth of 6 Mbps which results to a constraint that the bitrate shall be smaller or equal to the available bandwidth. In case of the the example from above the feasible adaptation parameters would be reduced to {1024→5, 640→3.5}. That is, the transmission of the full resolution {1902→20} would exceed the available bandwidth significantly but a resolution of 1024 pixel (or below) would not violate this constraint.

4. Based on the set of variables representing the adaptation parameters a large number of possible adaptation parameter combinations arise. Although the constraints might lead to parameter combinations that are not considered as valid (since they dissatisfy one or more constraints, i.e., there are infeasible adaptation parameters), it is still not guaranteed that the resulting set of valid decisions is unambiguous, e.g., contain only a single adaptation decision. For that reason the well-known concept of objective functions is used for the final selection of the adaptation decision among the set of feasible ones. An objective function consists of an algebraic expression which has to be minimized (or maximized).
Example: In the context of multimedia adaptation it might be useful to use a maximization constraint that selects those adaptation parameters that lead to the best frame rate, quality, resolution etc., depending on the user's preferences. As with the constraints, there may be more than one objective function. However, this can also lead to problems since the objective functions could be contradictory and require more sophisticated mathematical algorithms to handle.

The mathematical optimization problem can then be expressed as follows.

Let P be the set of all variables with the cardinality p. Each of the variables P_i has a domain D_i which is either a finite set of possible values or a closed interval of the real numbers. Let v be a vector of p values which represent a possible assignment of the variables, which means $v \in D_1 \times D_2 \times \ldots \times D_p$.
The set of all constraints is denoted by L and has the cardinality l. Each constraint denoted as L_j can be represented as a Boolean function $L_j(v) \rightarrow$ {true, false}. Furthermore, the set of objective functions can be modeled as a vector of objective functions O. Each of this vector's elements represents exactly one optimization constraint which has to be minimized. Note that, if the function should be maximized, it has to be transformed into a minimization function by multiplying it by -1.

The resulting optimization problem can be stated as follows.

minimize $O(v)$
subject to $L_j(v)$ = true for $j = 1 \ldots l$

Depending on the number of objective functions, three kinds of problems can be distinguished:

- If there are no objective functions defined, the problem will degrade to a *constraint satisfaction problem* [4]. In this case, each v that satisfies the constraints is a feasible solution for the problem.
- The second kind of problems arises when there is only a single objective function. The resulting mathematical problem is then called *single objective optimization problem*. The solution of the problem is the v that satisfies the limit constraints and minimizes the objective function.
- The third type of problem is called *multi-objective optimization problem* and is characterized by having more than one objective function [5]. As these functions can often be contradictory, a minimization of all functions cannot be achieved and a special treatment of them is necessary.

The solution of the optimization problems yields the optimum value assignment for the variables which does not violate any of the constraints and is optimal with respect to the objective function(s). The resulting adaptation decisions are simply the values of the variables that represent the adaptation parameters. The following example illustrates the concepts introduced above.

Example – Adaptation of Scalable Video Streams

An optimization problem that deals with the adaptation decision-taking of a scalable video stream – in our particular case of an MPEG/ITU-T Scalable Video Coding [6] stream – can be modeled as follows. Assume that the video is encoded in a way that the SVC base layer is encoded with a resolution of 640x360 pixels and two additional spatial enhancement layers exist which result in resolutions of 1024x576 and 1920x1080 pixels, respectively. Additionally, the video stream is assumed to be scalable along the temporal dimension. This means that, depending on the number of temporal layers, the framerate can be either 15 or 30 frames per second (fps). Following the idea of modeling the adaptation parameters as variables the spatial and temporal layers can be modeled as shown in **Table 1** (a).

Table 1. (a) Adaptation Parameter Variables; (b) Content Property Variables.

Variables	*Domain*
SpatLayer	{0, 1, 2}
TempLayer	{0,1}

Variables	*Domain*
HorizRes	[176, 1920]
VertRes	[144, 1080]
FrameRate	[1, 30]
BitRate	[0.5, 12]

Both variables are discrete since their domains are finite. The selection of the adaptation parameters has an impact on the properties of the content. As already introduced above, these properties have to be modeled as variables as well. The decision which properties and arising dependencies are modeled is depending on the actual use case. In the context of this example the relevant characteristics of the adapted video stream are the horizontal and vertical resolution (in pixels), the framerate (in fps) and the bitrate of the video (in Mbps) as shown in **Table 1** (b).

Following the optimization based approach the dependencies between the variables that represent both adaptation parameters and properties have to be modeled explicitly. In this example the relationships are quite obvious. The selection of the number of temporal layers has a direct influence on the framerate of the video. On the other hand the horizontal resolution is determined by the number of spatial layers. In contrast to that the resulting bitrate is influenced by both the number of temporal and spatial layers. All of the three dependencies have in common that the function is only valid for a finite number of arguments, which makes it possible to express the relationship by using a look-up table. For modeling the dependency between horizontal and vertical resolution it is more appropriate to use an algebraic expression. Since we assume that the aspect ratio of 16:9 is preserved, the vertical resolution is simply the horizontal resolution times 9 divided by 16. Formally, the functional dependencies can be given as shown in **Table 2**.

Table 2. Functional Dependencies.

Function	*Mapping*
HorizRes = f(SpatLayers)	{0→640, 1→1024, 2→1920}
VertRes = f(HorizRes)	VertRes = HorizRes / 16 * 9
BitRate = f(SpatLayers, TempLayers)	{(0,0)→2, (1,0)→5.5, (2,0)→7.5, (0,1)→3, (1,1)→8, (2,1)→12}
FrameRate = f(TempLayers)	{0→15, 1→30}

In the next step the constraints imposed by the usage context have to be modeled. Assume two limitations that are based on the available network bandwidth and the display capabilities of the user's end device. The available network bandwidth might limit the bitrate of the video to be lower than 7 Mbps since otherwise no streaming of the video is possible. The maximum screen resolution of the end user device might be 1024x768 pixels but the device is also capable to handle lower resolutions. Both constraints impose limitations on which values a variable can take. Formally, these constraints can be expressed like shown in **Table 3**.

Table 3. Limitation Constraints.

Name	*Syntax*
Horizontal Resolution	HorizRes ≤ 1024
Vertical Resolution	VertRes ≤ 768
Bitrate	BitRate ≤ 7

Finally, an objective function has to be specified that steers the selection of the final adaptation parameters. Assume that the video should be adapted in a way that it maximizes the framerate of the video in favor of maximizing the resolution. The objective function would then be *maximize FrameRate*.

The optimization problem can then be solved as follows. Since there is a restriction on the vertical and horizontal resolution, the parameter for the number of spatial layers has to be either 0 or 1 (2 would lead to a resolution of 1920x1080 pixels which would violate two constraints). The third limitation constraint that restricts the video bitrate to be lower than 7 Mbps further prevents the selection of the value 1 as parameter for both SpatLayer and TempLayer. This would lead to a video bitrate of 8 Mbps which would violate the constraint. After the processing of the limitation constraints only three combinations of adaptation parameters remain. Among these feasible parameters the one which maximizes the objective function is finally taken. In the context of this example the final parameter selection is SpatLayer = 0 and TempLayer = 1. This results in a video bitrate of 3 Mbps and a framerate of 30 fps.

Interoperability Support for Optimization-based Adaptation Decision-Taking

Part 7 of MPEG-21 – Digital Item Adaptation (DIA) – deals with the adaptation of multimedia content [7]. Since decision-taking plays a vital role for adaptation, it also defines descriptions that can be used for steering decision-taking. The approach intended within MPEG-21 is based on the optimization problem approach introduced above. XML-based metadata is used to define the optimization problem which actually incorporates the adaptation logic. The metadata itself can be interpreted by a component that determines the adaptation decision by solving the optimization problem. The advantage of this metadata-driven approach is that the adaptation decision-taking engine (ADTE) can remain generic since the logic is defined by the metadata. It should be pointed out that the MPEG-21 standard only defines the syntax and semantics of the descriptions, but does not cover the algorithms that have to be used for solving the optimization problem.

The MPEG-21 description formats for optimization-based adaptation decision-taking are

- Adaptation QoS (AQoS)
- Usage Environment Description (UED)
- Universal Constraint Description (UCD)

Adaptation QoS (AQoS) provides a way for expressing the adaptation capabilities of the multimedia content and the resulting content properties. It can be used for the first two modeling steps which include the definition of variables and their interrelationships. In MPEG-21 DIA terminology, variables are called IOPins. An IOPin has a unique name and can hold a value which must be contained in the variable's domain. IOPins can either be discrete or continuous. For defining the relationships between the IOPins, the standard offers three different mechanisms. Discrete functions can be defined by either a look-up table or a utility function. Although they are quite similar, the main difference is that utility functions are more suitable for specifying sparse functions, which means that the function value is defined only for a subset of input parameter value combinations. Relationships between continuous IOPins can be expressed by stack functions which represent algebraic expressions. The name stack function is based on the fact that the expressions are in postfix notation (Reverse Polish Notation), which can be evaluated easily by using a stack.

The *Usage Environment Description (UED)* specifies a normative way to describe a variety of properties related to the user and his/her usage context, thus supporting the vision of UMA. UEDs cover four major aspects which are relevant for adaptation decision-taking, namely:

- the user characteristics, which enable the exact description of the user with his/her preferences and impairments;
- the terminal capabilities, that can be utilized for describing capabilities and limitations related to the user's end device;
- the network characteristics, which are imported when considering adaptations of streamed multimedia resources; and
- the natural environment characteristics, which can be used to further define the surrounding area of the user that consumes multimedia content.

The characteristics defined in a UED, e.g., the resolution of the terminal's screen or the available network bandwidth in bps, can be referenced when defining constraints that limit the values of the IOPins.

The *Universal Constraints Description (UCD)* defines the syntax and semantics for declaring additional constraints for the adaptation and can be seen as the linking element between the Adaptation QoS and the Usage Environment Description. On the one hand it can be used to restrict the values of IOPins based on metadata from the UED. This restriction is achieved by stating the constraints as mathematical functions called *limit constraint* in MPEG-21 DIA terminology. On the other hand it introduces *optimization constraints* that allow defining preferences concerning the adaptation operations. Optimization constraints are basically the objective functions of the optimization problem.

Based on these three types of descriptions, the ADTE can extract the optimization problem and determine the adaptation decision by solving the optimization problem. However, one of the drawbacks of the MPEG-21 approach is that it does not specify a dedicated algorithm for the solution of such problems. As the optimization problems that can be constructed by using this metadata falls into a very generic class of optimization problems, there exists no optimization algorithm that is complete. An algorithm is complete, if it finds a solution if one exists and otherwise correctly reports that no solution is possible. This problem can be leveraged by considering only discrete variables. In that case there exist a finite number of combinations that can be explored by an exhaustive search in the problem space. Although the problem space increases exponentially with the number of variables, the number of variable combinations to investigate for adaptation decisions is typically on the order of some hundreds to a few thousands. Another disadvantage of the MPEG-21 DIA approach is that codec selection problems (e.g., select one of the codecs supported by the user's end device) cannot be modeled appropriately.

Concluding Remarks

Optimization-based adaptation decision-taking enables one to configure the actual multimedia content adaptation engines in a way that optimizes the multimedia content's properties according to the usage environment at hand and the content. This approach has been adopted by the MPEG-21 DIA standard in a generic way and some open issues have been highlighted in this section.I It is also open to some extent how this approach shall interact or can be integrated with the other techniques, e.g., [8], in the field of adaptation decision-taking.

See: Adaptation Decision-Taking, Knowledge-based Multimedia Adaptation Decision-Taking, MPEG-21 Digital Item Adaptation, MPEG-21 Multimedia Framework

References

1. A. Vetro, C. Christopoulos and T. Ebrahami, Eds., "Special Issue on Universal Multimedia Access", *IEEE Signal Processing Magazine*, vol. 20, no. 2, March 2003.
2. 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, pp. 454-462, June 2005.
3. I. Kofler, C. Timmerer, H. Hellwagner, A. Hutter, and F. Sanahuja, "Efficient MPEG-21-based Adaptation Decision-Taking for Scalable Multimedia Content", *Proceedings of the 14th SPIE Annual Electronic Imaging Conference – Multimedia Computing and Networking (MMCN 2007)*, San Jose, CA, USA, January/February 2007.
4. Edward Tsang, *Foundations of Constraint Satisfaction*, Academic Press, 1993.
5. Y. Sawaragi and H. Nakayama and T. Tanino, *Theory of multiobjective optimization*, Academic Press, 1985.
6. H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103-1120, September 2007.
7. A. Vetro, C. Timmerer, "Digital Item Adaptation: Overview of Standardization and Research Activities", *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 418–426, June 2005.
8. D. Jannach, K. Leopold, C. Timmerer, and H. Hellwagner, "A Knowledge-based Framework for Multimedia Adaptation", *Applied Intelligence*, vol. 24, no. 2, pp. 109-125, April 2006.