# Utility Function for Semantic Video Content Adaptation

Vanessa EL-KHOURY, Nadia BENNANI
Lyon University, CNRS
INSA-Lyon, LIRIS, UMR5205
F-69621, France
{firstname.lastname}@insa-lyon.fr

David COQUIL
Passau University
Chair of Distributed Information Systems
94032 Passau, Germany
{firstname.lastname}@uni-passau.de

## ABSTRACT

The vision of Universal Multimedia Access (UMA) and Universal Multimedia Experience (UME) has driven research in the multimedia community for a long time. Implementing content adaptation frameworks to satisfy heterogeneous types of constraints is among the main requirements for UMA and UME. At the core of these frameworks lies the adaptation decision engine, which computes the appropriate adaptation plans. Though much research has already been done in this domain, the problem of semantic constraints has generally been neglected. This paper addresses the problem of selecting the optimal adaptation operation to satisfy semantic constraints while maximizing the utility of the adapted video. To this end, we define a utility function that computes a value for each possible adaptation operation. We represent our utility function using the MPEG-21 Digital Item Adaptation (DIA) tools. This facilitates the integration of semantic constraints with other types of constraints in MPEG-21 Universal Constraint Description format.

## Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems– *video*.

## Keywords

Universal Multimedia Experience, Semantic Adaptation, MPEG-21 DIA, Utility Function.

## 1. INTRODUCTION

Nowadays, users access Internet services such as video consumption services using a large variety of devices (PDA, PC, Mobile phone, etc.) with different capabilities (computational power, memory size, display size, etc.). In addition, these users have different requirements that vary according to several dimensions: profile (age, mood, interests, etc.), preferences (grayscale color, video format, etc.), network capabilities (bandwidth, bitrates, etc.). Given this heterogeneity, a content adaptation engine is required to provide an adapted content that satisfies this whole range of users. Ultimately, the adaptation enabled by this engine would realize the visions of *Universal Multimedia Access* (UMA) and *Universal Multimedia Experience* (UME) [1]. UMA focuses on adapting the multimedia content to meet technical limitations such as user terminal or network, whereas UME concentrates on adapting the content to meet semantic constraint ensuring a consistent informative adapted content to the user. Thus, the primary difference between these two notions is that UME recognizes the user instead of the terminal as the end point of universal multimedia consumption.

Many adaptation frameworks were presented in the literature targeting UMA and UME [2] [3]. At the core of these frameworks lies the adaptation decision-taking engine (ADTE), which computes the appropriate adaptation plans to be executed on the multimedia content. In [2], the authors present CAIN-21, a multimedia adaptation engine that aims to automate interoperability among multimedia formats and systems. To this end, a set of well described adaptation tools called Content Adaptation Tools (CATs) are described integrating different content adaptation approaches: transcoding, transmoding, scalable content, and temporal summarization. A multi-step adaptation is used enabling the combination and execution in several steps of the CATs. Afterwards, a quality-based method is executed to identify the conversion sequence that yields maximum quality. Davy et al. [3] present NinSuna, a format-independent multimedia content adaptation and delivery platform based on a model for multimedia bitstreams. It supports the exploitation of scalability to meet usage environment, as well as semantic adaptation to meet user preferences. This model makes use of semantic Web technologies to take semantic decisions. However, the adaptation decision making method is not yet mature. The client implicitly indicates the desired coding and delivery format information and the ADTE simply matches it with the scalability information of the requested content to compute the adaptation parameters.

The adaptation engines of CAIN-21 and NinSuna target UMA and UME by adapting multimedia content according to the usage environment and user preferences. However, they do not completely cover the requirements of UME. In particular, they can neither adapt the content at the object level nor maintain the semantic integrity of the adapted video. This prevents them to deal with several types of semantic constraints (e.g.; no cigarette). This paper proposes an adaptation engine that satisfies semantic constraints targeting the UME. In particular, we tackle the critical process of computing the adaptation plan to deliver the best possible adaptation in terms of *Information* and *Quality* to the end user. To this end, we define a utility function similar to the one presented in the MPEG-21 AdaptationQoS tool [4]. Doing so will enable the integration of semantic constraints in the general MPEG-21 DIA framework along other types of constraints. In fact, UF describes the trade-off relationship between resources and utilities along each adaptation dimension. It plays a key role in choosing the optimal adaptation among multiple options that meet resource constraints or user preferences. In this paper, we present a utility function that computes utility values for adaptation operations applied to video segments identified during a preliminary process. Thus, we aim to define a piecewise adaptation framework. The utility function makes use of metadata provided by MPEG-7 description tools [5] that describe the content of the video in order to compute the impact of possible adaptation operations along several dimensions: affected area, impact on priority segments and resultant visual coherence.

The remainder of this paper is organized as follows. Section 2 presents the preliminary processes that precede the application of the utility function. In Section 3, we introduce necessary basic definitions and Section 4 describes the utility function and its parameters. In Section 5, we present our system prototype for the semantic adaptation of the video content. Finally, conclusion and future work are given in Section 6.

## 2. PRELIMINARY PROCESSES

In this section, we describe the processes that must take place before the utility function described in this paper is applied. These processes produce the inputs needed by our function.

1. *Video segmentation*; in this paper, we propose a piecewise adaptation approach. Indeed, contrarily to the case of technical constraints in which the same adaptation operation is applied to the whole video; it may be more appropriate for semantic adaptation to apply distinct operations to different segments of the video. Thus, the video should be first segmented into entities, which are the basic units of the adaptation process.

2. *Video annotation*; we assume that MPEG-7 annotation tools [6] have been used to produce a description of the content. This annotation must contain all the information needed by the utility function to evaluate the impact of the adaptation operations. First, the set of frame intervals in which the objects of interest appear must be specified; in the remainder of the paper, we will restrict ourselves to semantic constraints in which a set of objects should be excluded from the adapted video. Then, for each frame within the intervals, we need the position and the size of the target objects. This can be provided by object tracking techniques. Finally, we need a specification of priority segments of the video. *Priority* here has a semantic meaning, corresponding to key scenes without which the video could not be understood. This information will be provided by a human annotator. To represent it, we can use the MPEG-7 *Priority* attribute from the *VariationSetDS* description scheme. As depicted in *Figure 1*, our priority value denoted $\rho$ takes two values: *1* if the content of the video segment is important and *0* by default.

3. *Constraint processing*; the starting point of the adaptation process in our framework is what we call a *generic semantic constraint* (e.g.; no publicity). To adapt a specific video, the generic constraints must be pre-processed to derive the *instantiated semantic constraints* (e.g.; the soda bottle shown in frames 24 to 42 should be excluded). We assume that this process, which uses the MPEG-7 annotation and external knowledge, has occurred beforehand.

## 3. BASIC DEFINITIONS

**Definition 1 (Entity)**

An entity $e$ is the basic unit of video that undergoes the adaptation process. In an adaptation framework, entities may exist at different levels, such as pixels, objects, frames, shots, scenes, sequences, etc. In our semantic adaptation framework, we define the entity as a segment of consecutive frames sharing semantically the same meaning. An entity may correspond to a shot that has been automatically identified by shot detection techniques. However, a shot identified by these techniques may contain several segments with different semantic meanings. In this case, the shot will be divided into sub-shots by a human annotator, and the entities used by the adaptation framework will be the sub-shots (*Fig.1*). Based on this, we formally define a video $V=\{e_k;\ k=1:n\}$ as a set of such entities $e$, $n$ being the total number of entities of the video.

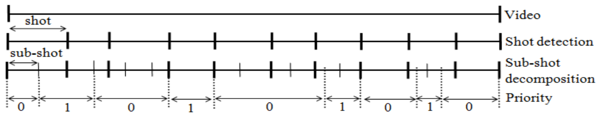

**Figure 1. Video entity definition with priority value.**

**Definition 2 (Generic semantic constraint)**

A generic semantic constraint $CT$ is a restriction specifying a category of objects that should be excluded from the adapted video (e.g.; no publicity).

**Definition 3 (Instantiated semantic constraint)**

An instantiated semantic constraint $ct$ is a restriction bound to one salient object of a video derived from a generic semantic constraint (e.g.; no soda bottle). It is defined by an object and a set of frame intervals called a *Group of Frames* (GOF).

The instantiation process determines the set of salient objects $\{o_i\}$ satisfying a generic semantic constraint and creates exactly one instantiated constraint $ct_i$ for each $o_i$. Then for each $ct_i$ and each entity $e_k$ in which $o_i$ appears, it identifies the *GOF* denoted $GOF_{ek,oi}=\{[f_i,f_j]\}$ where the salient object $o_i$ appears (*Fig. 2*). As the MPEG-7 description also provides information about the position of the object in a frame, for each frame in the intervals defined in *GOF*, the position and the size information of the object in the frame are known.

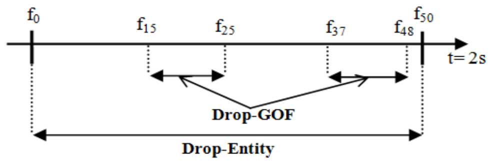

**Figure 2. Constraint instantiation and adaptation operations description.**

**Definition 4 (Adaptation space)**

The adaptation space represents all the possible adaptation operations $A=\{a_i\}$, such as $a_i$ is one of the adaptation operators of the finite set *{Drop_Entity, Drop_GOF, Drop_Object}*. For simplicity, we do not yet consider the case of applying different operations to different intervals of a GOF; we define *Drop_GOF* as dropping all intervals of the *GOF* and *Drop_Object* as removing the salient object from all frames of the *GOF (Fig. 2)*.

**Definition 5 (Utility)**

As defined in MPEG-21 DIA [2], the utility $u$ is a measurement of the quality of the video resulting from an adaptation operation. This value is computed by a utility function depending on several parameters (see next section).

## 4. UTILITY FUNCTION

The use of standards to describe usage context information enables to achieve a high degree of interoperability and to support a wide range of devices. Thus, in our framework we use MPEG-7 to represent the video content description and MPEG-21's *AdaptationQoS* tool [4], which provides an interoperable format for describing a utility function. Using this tool, the relation between (Constraints, Adaptation operators, Utility) can be represented and provided to the adaptation engine to let it select the optimal adaptation plan.

## 4.1 Adaptation Problem

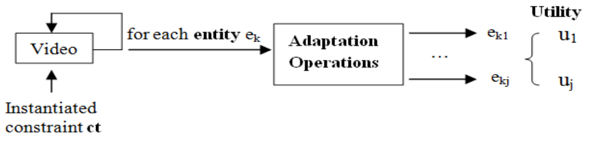

**Figure 3. Conceptual framework for semantic video content adaptation.**

As shown in *Figure 3*, each adaptation operation $a_i$ yields an adapted entity $e_{kj}$ having a utility value $u_j$. Thus, we formally formulate the adaptation problem as follows:
*Given a content* ***entity*** *and an* ***instantiated semantic constraint****, select the* ***optimal adaptation operation*** *so that the* ***utility*** *of the adapted entity is maximized.*

## 4.2 Utility Function Representation

To resolve the problem described above, we define a utility function. Its purpose is to evaluate the impact of an adaptation operation on an entity according to several parameters and to aggregate these values in a single one. This enables a quantitative comparison between different adaptation operations.
The function is formally defined as follows: given a set of possible adaptation operations $A=\{a_i\}$, an instantiated constraint *ct* and an entity *e*, the utility function (*UF*) computes for each $a_i$ a value $v_{ai}$, such that the larger this value the higher the utility of selecting $a_i$ to adapt *e* for *ct*. The utility function is defined as:

$$UF = 1 - f(p_1, \dots, p_k)$$

Where *k* denotes the number of parameters used in the utility function, each parameter being normalized to vary between *0* and *1*. As a first stage, we define the $f(p_1,\dots,p_k)$ as the weighted means of the data set $\{p_1,\dots,p_k\}$ with non-negative weights $\{w_1,\dots,w_k\}$ as follows:

$$f(p_1, \dots, p_k) = \frac{\sum_{i=1}^{k}(w_i p_i)}{\sum_{i=1}^{k} w_i}$$

And as the weights are normalized such that $\sum_{i=1}^{k} w_i = 1$:

$$UF = 1 - \sum_{i=1}^{k}(w_i p_i)$$

Based on the above information, we detail in the next section the utility function by describing the parameters as well as the formulas that compute them for each adaptation operation.

## 4.3 Parameters Definition

Many parameters should be considered to evaluate the utility of video that has been adapted to satisfy a semantic constraint. At the physical level, we can examine the rate of the area affected by the adaptation, as well as the adaptation processing cost that varies from an adaptation operation to another. At the semantic level, we can study the loss of information due to the number of the priority frames influenced by the adaptation. Moreover, since the loss of information may lead to an inconsistency of the entity content, it is important to consider the visual and semantic coherence of the resulting video. The first version of the utility function described in this paper, uses the following parameters: (1) the affected area, (2) the affected priority area, (3) the visual coherence.

### *4.3.1 Affected Area*

When semantically adapting entity content, we must try to minimize the area affected by the adaptation. This area varies from an adaptation operation to another. The role of the parameter $p_1$ is to evaluate the surface of this area.
For the *Drop_GOF* operation, $p_1(Drop\_GOF)$ is defined as:

$$p_1(Drop\_GOF) = \frac{\sum_{i=1}^{n} l_i}{N}$$

Where *N* is the number of frames in the entity, *n* is the number of frame intervals of the *GOF* satisfying a constraint *ct* within the entity and $l_i$ is the number of frames in the $i^{th}$ frame interval. According to this definition, the value of $p_1(Drop\_Entity)$ is obviously *1*. For the *Drop_Object* operation, we calculate for each frame containing the salient object *o*, the ratio of the object size $size_o$ over the frame size $size_f$ .Thus, $p_1(Drop\_Object)$ is given by:

$$p_1(Drop\_Object) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{l_i} size_{oij}}{N * size_f}$$

Where $o_{ij}$ is a salient object of the frame situated at the position *j* in the $i^{th}$ frame interval of a *GOF* within an entity.
To summarize, for an adaptation operation $a_i$, the function $p_1(a_i)$ is determined as follows:

$$p_1(a_i) = \begin{cases} 1 & \text{if } a_i = Drop\_Entity \\ \dfrac{\sum_{i=1}^{n} l_i}{N} & \text{if } a_i = Drop\_GOF \\ \dfrac{\sum_{i=1}^{n}\sum_{j=1}^{l_i} size_{oij}}{N * size_f} & \text{if } a_i = Drop\_Object \end{cases}$$

### *4.3.2 Affected priority area*

Besides quality degradation in the user perception, adapting priority frames causes loss of information leading to an inconsistency of the adapted entity. For instance, a *Drop_GOF* operation may affect a large area (high $p_1$ value) without impacting priority frames. Therefore, it is also important to study the surface of the affected priority area for each adaptation operation $a_i$. Let $p_2$ be the function related to this parameter. For the *Drop_GOF* operation, $p_2(Drop\_GOF)$ is given by:

$$p_2(Drop\_GOF) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{l_i} \rho_{ij}}{\sum_{k=1}^{N} \rho_k}$$

Where $\rho_{ij}$ is the priority value of the frame at position *j* in the $i^{th}$ frame interval of a *GOF* within an entity, $\rho_k$ is the priority value of the $k^{th}$ frame in the entity and *N* is the total number of frames in the entity. Similar to $p_1$, the value of $p_2(Drop\_Entity)$ is *1*. For *Drop_Object*, $p_2$ computes the affected priority area based on the ratio between the object size and the frame size. In this case, $p_2(Drop\_Object)$ is defined as follows:

$$p_2(Drop\_Object) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{l_i} s_{ij}}{\sum_{k=1}^{N} \rho_k}$$

Where $s_{ij}$ is the object-priority-frame-dropping factor of an object *o* in a frame at a position *j* in the $i^{th}$ interval of the *GOF* with a priority degree $\rho_{ij}$ given by:

$$s_{ij} = \rho_{ij}\frac{size_{oij}}{size_f}$$

Therefore, for an adaptation operation $a_i$, the function $p_2(a_i)$ is determined as follows:

$$p_2(a_i) = \begin{cases} 1 & \text{if } a_i = Drop\_Entity \\ \dfrac{\sum_{i=1}^{n}\sum_{j=1}^{l_i}\rho_{ij}}{\sum_{k=1}^{N}\rho_k} & \text{if } a_i = Drop\_GOF \\ \dfrac{\sum_{i=1}^{n}\sum_{j=1}^{l_i}s_{ij}}{\sum_{k=1}^{N}\rho_k} & \text{if } a_i = Drop\_Object \end{cases}$$

### 4.3.3 Visual Coherence

It is difficult to come up with an adequate visual coherence definition as it is a subjective notion which perception differs from user to user. In our case, we measured it in term of *Gaps* resulting from the adaptation when removing group of frames. Indeed, even if the video is re-encoded, the user may notice that many frame intervals of large size have been removed from the original video. Thus, the visual coherence can be measured by a parameter $p_3$ depending on the size and frequency of the gaps. For the *Drop_GOF* operation, $p_3(Drop\_GOF)$ is given by:

$$p_3(Drop\_GOF) = \frac{\sum_{i=1}^{n}(1 - exp(-\frac{size_{gap}}{\sigma}))}{n}$$

Where $size_{gap}$ is the ratio of the size of the *GOF* over the size of the entity and $\sigma$ is a threshold (to be experimentally defined later) such that a gap size above $\sigma$ creates a noticeable inconsistency.
For *Drop_Object*, the value of $p_3$ is *0* as it does not cause a frame dropping. The *Drop_Entity* operation creates a single gap of the size of the entity. Thus, $p_3(Drop\_Entity)$ is given as follows:

$$p_3(Drop\_Entity) = \frac{1 - exp(-\frac{1}{\sigma})}{n}$$

Below, we describe the function $\boldsymbol{p_3(a_i)}$ for each of the adaptation operations $\mathbf{a}_i$:

$$p_3(a_i) = \begin{cases} \dfrac{1 - exp(-\frac{1}{\sigma})}{n} & \text{if } a_i = Drop\_Entity \\ \dfrac{\sum_{i=1}^{n}(1 - exp(-\frac{l_i/N}{\sigma}))}{n} & \text{if } a_i = Drop\_GOF \\ 0 & \text{if } a_i = Drop\_Object \end{cases}$$

$N$ is the total number of frames in an entity, $L$ is the length (frames) of a video, $n$ is the frame interval number of the *GOF* and $l_i$ is the number of frames in the $i^{th}$ frame interval.

## 5. SYSTEM PROTOTYPE

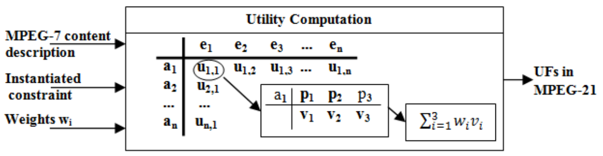

**Figure 4. Utility computation system.**

A utility function framework for computing the utility values, depicted in *Figure 4,* is under development. The system takes as input the weights, an instantiated constraint and an MPEG-7 document describing a video. For each entity and each possible adaptation operation, it parses the MPEG-7 document to extract the necessary information and computes the parameters using the formulas defined above. The parameters are then aggregated into a utility value using the weights specified by the user. As an output, a *UF* in a MPEG-21 format is generated.

## 6. CONCLUSION

In my thesis, I address the problem of choosing an adaptation operation for video content to satisfy semantic constraints while maintaining the semantic integrity of the content. In this paper, we have introduced a function that measures the utility of the adapted entity produced by an adaptation operation. This function is based on the framework defined in MPEG-21 DIA. Its current version depends on three parameters: the affected area, the affected priority area and the visual coherence. These parameters are computed using information extracted from an MPEG-7 description of the video content, and aggregated into a utility value. This enables a numerical comparison among possible adaptation operations for each entity composing the video.
One limitation of the current version of the utility function is that its parameters are all favorable to *Drop_Object*. Thus, the next step is to improve function by introducing other parameters, which will penalize this operation. The first parameter will evaluate the processing cost of the operations; the second new parameter will evaluate the visual quality degradation introduced by *Drop_Object*. This will be facilitated by the properties of the utility function that has been designed to be easily extensible with additional parameters. I also plan to make the framework more complete by considering the possibility of applying distinct operations to different intervals of the same entity and by adding *Blur_Object* to the set of defined operations. In addition, I plan to treat the case where the adaptation of an entity deals with more than one semantic constraint. Finally, as our semantic adaptation proposal is defined to make it compatible with MPEG-21, an important direction of future work is to design and implement a global video adaptation framework capable of simultaneously and consistently processing semantic and technical constraints.

## 7. REFERENCES

[1] Pereira, F., Burnett, I. 2003. Universal multimedia experiences for tomorrow. IEEE Signal Processing Magazine, vol. 20, no. 2, Portugal, Mar. 2003, pp. 63-73,.

[2] López, F., Martínez, J. M., and García, N. 2009. CAIN-21: An Extensible and Metadata-Driven Multimedia Adaptation Engine in the MPEG-21 Framework. SAMT 2009: 114-125.

[3] Deursen, D., Lancker, W., Neve, W., Paridaens, T., Mannens, E., and Walle, R. NinSuna: a fully integrated platform for format-independent multimedia content adaptation and delivery using Semantic Web technologies. Multimedia Tools Appl. 46, 2-3, Jan. 2010, pp. 371-398.

[4] ISO/IEC 21000-7:2004, Information Technology - Multimedia Framework (MPEG-21)-Part7: Digital Item Adaptation, 2004.

[5] Martínez, J. M. MPEG-7 Overview (version 10). ISO/IEC JTC1/SC29/WG11/N5525, Palma de Mallorca, Oct. 2004.

[6] Döller, M. and Lefin, L. 2007. Evaluation of available MPEG-7 Annotation Tools. In Proceedings of I-MEDIA /I-SEMANTICS '07, Graz, Austria, Sept. 2007.