Automatic adaptation decision making in the MPEG-21 framework: mechanisms and complementary description tools*

Fernando López[1], José M. Martínez[1], Narciso García[2]
[1]VPULab, EPS – Universidad Autónoma de Madrid, {f.lopez,josem.martinez}@uam.es
[2]GTI, ETSIT – Universidad Politécnica de Madrid, narciso@gti.ssr.upm.es

Abstract

This paper explains an effective mechanism to make automatic multimedia adaptation decisions within the MPEG-21 framework. The paper analyzes some difficulties for the implementation of automatic decision with the current MPEG-21 description schema. Subsequently, the paper proposes some improvements to the MPEG-21 description schema to address these difficulties. To demonstrate these improvements, the current implementation of the CAIN-21 framework is explained and several experiments are reported.

1. Introduction

Multimedia adaptation offers content providers added value by increasing the range of terminals that consume their contents. In addition, it improves the quality of such content by offering the content provider and its customers the capability to customize the content to their individual preferences. The MPEG-21 [1] framework has standardized the representation of the elements that comprises a multimedia system.

This paper describes a mechanism that enables the automatic selection of a group of conversions that adapt an MPEG-21 multimedia resource to the constraints of the usage environment (terminal capabilities, network constraints and user's preferences). The paper also describes how this mechanism has been integrated and tested in the CAIN-21[1] [2] adaptation engine. As a result, this paper identifies a set of limitations in the description capabilities of the MPEG-21 framework and proposes some extensions to the MPEG-21 description schema to remove this ambiguity.

Section 2 introduces the MPEG-21 description schema and the elements involved in the adaptation. Section 3 explains the decision algorithm and the software used for the experiments. Section 4 identifies several semantic gaps and ambiguities in the decision mechanism and some extensions to the current MPEG-21 schema are proposed. Section 5 demonstrates how these extensions make possible to address a new range of adaptation problems. Section 6 concludes the paper.

2. Related Work

Within the MPEG-21 framework [1], multimedia elements are represented as MPEG-21 Part-2 *Digital Items* (DIs). According to this framework, a DI is a multimedia container representing one or more *Component* elements. Each *Component*[2] element conveys *resources* (media files) and *descriptors* (metadata). Usually, this metadata is represented by means of one or more MPEG-7 descriptions [3].

MPEG-21 Part-7 uses the term *Digital Item Adaptation* tools (*DIA* tools) to refer to description tools collecting information related to the adaptation process. Instances of the *DIA* tools are referred to as *DIA* descriptions. The *DIA* tools enable the description of two main actors: the *content provider* and the *content consumer*. Specifically, the content provider's media streaming and quality of service are described by means of the *AdaptationQoS* tools and the *Universal Constraints Description* tools (*UCD* tools). The *Usage Environment Description* tools (*UED* tools) describe the content consumer, particularly in terms of the terminal capabilities, network conditions, users' preferences and natural environment. Instances of the *UED* tools are referred to as *UED* descriptions, which are classified into four groups: *Terminals*, *Networks*, *Users* and *NaturalEnvironments*.

CAIN-21 [2] has proposed the notion of *Component Adaptation Tools* (CATs) to represent pluggable software tools capable of performing one or more *conversions* over a specific MPEG-21 *Component*. To enable the automatic selection of the group of conversions that adapts the *Component* elements of a DI to the existing *UED* descriptions, the underlying algorithm must be

* Work financed by the Spanish Government (TEC2007-65400 - SemanticVideo) and the Ministerio de Educación y Ciencia of the Spanish Government through the FPU fellowship grant under the name of the first author.
[1] The CAIN-21 software along with a CAIN-21 demo are publicly available at *http://cain21.sourceforge.net*
[2] MPEG-21 capitalizes and italicises XML description elements. This paper adopts this rule.

informed of the adaptation capabilities for each conversion. The *CAT Capabilities* is the description elements that CAIN-21 uses to provide this information.

3. Decision mechanism

3.1. Multimedia properties

CAIN-21 introduced an addressing mechanism [4] in which changes in the metadata descriptors will not imply changes within the underlying source code. With this mechanism, metadata is dealt by *properties*. The *Properties DI* is intended to store all keys and corresponding XPath/XPointer expressions. The properties of the multimedia content and the *CAT Capabilities* are described using MPEG-7 descriptions. The most relevant properties from the standpoint of this research are summarized in Table 1. In the same manner, the most relevant MPEG-21 properties of the *UED* description are summarized in Table 2. The entire set of properties and their XPath/XPointer expressions are available in the CAIN-21 software.

name	Description
url	Resource file
content	Modality of the resource: *Video*, *Audio* or *Image*.
format	System format of the resource
bitrate	Bitrate of the system resource
visual_format	Visual stream format of the resource.
visual_bitstream	Visual stream bitrate
audio_format	Audio coding format of the resource.
audio_bitrate	Audio stream bitrate

Table 1: Properties of the multimedia content (MPEG-7)

name	Description
binding	Delivery mechanism: *file*, *http*, *rtp*
transport_decoding_format	Container formats supported by the terminal
image_decoding_format	Image formats supported by the terminal
visual_decoding_format	Visual formats supported by the terminal
visual_bitrate	Maximum bitrate in the visual stream
audio_decoding_format	Audio formats supported by the terminal
audio_bitrate	Maximum bitrate in the audio stream

Table 2: Properties of the terminal (MPEG-21)

3.2. The matching process

Knowledge-based decision algorithms identify multi-step sequences of conversions that adapt multimedia to the usage environment [5]. CAIN-21 uses a knowledge-based decision algorithm [6] to identify the sequence of conversions that adapts a *Component* to the *UED*. CAIN-21 has divided in [2] the decision mechanisms into *static decisions* (which do not require access to the media resource) and *dynamic decisions* (which require access to the media resource). The content provider's constraints (*AdaptationQoS* and *UCD*) are dynamic decisions, and therefore these tools are not considered during the matching process. However, as explained in [2], they may be used during the dynamic decisions that take place throughout the execution of the CATs.

In brief, the matching process, theoretically explained in [6], represents the properties of the terminal as constraints. The decision algorithm implements a planner that identifies a sequence of conversions (implemented in the CATs). This sequence of conversion transforms the properties of the DI to correspond to the properties of the terminal. For example, if the *Component* of the DI has the following properties:

format={*MPEG*}
visual_format={*mpeg2video*}
audio_format ={*mp3*}

The terminal has these properties:

visual_decoding_format={*h263*, *h264*}
audio_decoding_format={*mp3*,*aac*}

The *VideoTranscoderCAT* defines (within its *CAT Capabilities*) the following preconditions:

format={MPEG, MP4, AVI}
visual_format={*mpeg1video*, *mpeg2video*}
audio_format ={*wav*,*mp3*,*aac*}

and with the following postconditions:

format={*MPEG*, *MP4*}
visual_format={*mpeg2video*,*h264*}
audio_format ={*wav*,*mp3*,*acc*}

In this case, the decision algorithm of CAIN-21 automatically determines that the adaptation can be conducted in one step with the following *output parameters*, this being the subset of the postconditions that drive the adaptation:

format={*MP4*}
visual_format={*h264*}
audio_format ={*mp3*}

Please, refer to the public demo of CAIN-21 for a comprehensive set of examples demonstrating a wider set of properties. This demo includes experiments, in which the matching process automatically computes multi-step sequences of conversions.

4. Limitations and proposals for extension

4.1. Semantic gaps between MPEG-7/21

While implementing the matching process, described in Subsection 3.2, semantic gaps were identified between the MPEG-7 Part 5 properties (described with a *Component*) and the MPEG-21 Part 7 properties (described with *UED* descriptions):

1) The *mpeg7:Content* description element indicates the modality of the multimedia content (images, video, audiovisual, audio, etc). This property is used both in the description of the *Component* and in the postconditions of the *CAT Capabilities*. The *mpeg7:ContentCS* classification scheme standardises the available values for this element. The *content* property in Table 1 corresponds to this value. The standard *mpeg21:TerminalType*, however, does not includes any reference to the modality of the content that the terminal consumes.

2) The standard *mpeg21:TerminalType* does not provide any description of the *binding mode*, i.e., the delivery mechanism used to consume content (such as *file*, *http* or *rtp*).

3) The standard *mpeg21:TerminalType* does not specify if the properties of the terminal are mandatory or optional. For instance, if a terminal is defined by the *mpeg21:AudioCapabilitiesType*, does it mean that the adapted media must include audio? Or does it mean that this audio format could be consumed if present?

4.2. Inferred properties and ambiguity

Some missing properties in the MPEG-21 Part-7 *UED* can be inferred from existing properties. Others give rise to ambiguities. To remove these ambiguities, the properties must be described in the *UED*. Specifically:

1) The media content (e,g, image, video, audiovisual, audio) can be inferred from the standard *mpeg21:TransportCapabilitiesType*.

2) The binding modes of the terminal cannot be inferred from the standard *UED* description. Media can be delivered to the terminal using different delivery mechanisms (such as *file*, *http*, or *rtp*) whilst the terminal may not necessarily support all of them.

3) Optimal and mandatory properties of the terminal cannot be inferred if this information is not provided in the terminal description.

Therefore, these semantic gaps include both properties that can be inferred and properties that cannot be inferred (ambiguities). In particular, in the first gap the media content can be inferred from the *mpeg21:TransportCapabilitiesType* (illustrated in Listing 1). The second and third gaps demand an extension of the standard *mpeg21:TerminalType* and are addressed in the next subsection.

4.3. Proposal for extension

To address the previous semantic gaps, this paper proposes to extend the current *mpeg21:TerminalType*. Listing 1 shows a portion of the description of the terminal with *id="iphone"* from the CAIN-21 demo. The extensions that this subsection discussed are marked in bold. The XML Schema with these changes is publicly available in the file *cde.xsd* of the CAIN-21 software.

```
<Terminal id="iphone" xsi:type="cde:TerminalType">
 <TerminalCapability
    xsi:type="cde:HandlerCapabilitiesType">
  <Handler handlerURI="urn:mpeg:mpeg21:2007:01-BBL-
NS:handler:FILE"/>
 </TerminalCapability>
 <TerminalCapability
    xsi:type="cde:CodecCapabilitiesType">
  <cde:Decoding
     xsi:type="cde:TransportCapabilitiesType">
   <cde:Format
     href="urn:vpu:cs:FileFormatCS:2009:3gpp">
     ...........
   </cde:Format>
  </cde:Decoding>
  <cde:Decoding
     xsi:type="cde:VideoCapabilitiesType">
   <cde:Format
     href="urn:vpu:cs:VisualCodingFormatCS:2007:1">
    <mpeg7:Name xml:lang="en">
     H.264 Baseline Profile @ Level 1.1
    </mpeg7:Name>
   </cde:Format>
   <cde:CodecParameter
       xsi:type="CodecParameterBitRateType">
    <BitRate >32000</BitRate>
   </cde:CodecParameter>
  </cde:Decoding>
  <cde:Decoding xsi:type="cde:AudioCapabilitiesType"
              optional="true">
   <cde:Format href=
 "urn:mpeg:mpeg7:cs:AudioCodingFormatCS:2001:4.3.1">
    <mpeg7:Name xml:lang="en">
     MPEG-2 Audio AAC Low Complexity Profile
    </mpeg7:Name>
   </cde:Format>
   <cde:CodecParameter
     xsi:type="CodecParameterBitRateType">
    <BitRate>7950</BitRate>
   </cde:CodecParameter>
  </cde:Decoding>
 </TerminalCapability>
  ....................
</Terminal>
```

Listing 1: Extended *TerminalType*

In particular, this paper proposes two extensions to the standard *mpeg21:TerminalType*:

1) The delivery mechanism is fundamental in producing a proper adaptation as different CATs may enable different delivering mechanisms. MPEG-21 Part-18 [7] has proposed several *Bitstream Binding Languages* (BBLs) to describe the delivery mechanism. This paper proposes to reference the *mpeg18:Handler* description element of the BBLs inside the

cde:HandlerCapabilitiesType description element of the terminal (see Listing 1).

2). To describe whether a capability of the terminal is mandatory or optional (hard constraints or soft constraints according to [6]) further description is necessary. This paper proposes the *optional* attribute to provide this description. In Listing 1, the *cde:AudioCapabilitiesType* element includes the *optional* attribute in order to signal that the audio bitstream is not mandatory for the adapted content. If this attribute is absent, CAIN-21 considers the terminal property a mandatory constraint.

5. Experiments and validation

Several experiments, which are publicly available in the CAIN-21 demo, validate the proposed extensions.

The first experiment is the adaptation of a DI with an image (named *castle.xml*) to a video terminal named *iphone* (shown in Listing 1). CAIN-21 produces the following sequence of conversions *initial* → *image_transcoder* → *image_2_video* → *video_transcoder* → *goal*. In this sequence *initial* represents the original DI. The *image_transcoder* conversion transcodes the image format and size to the preconditions of the *image_to_video* conversion (i.e., JPEG image format and 3:4 aspect ratio). The *image_to_video* conversion can only produce MPEG-2 video and the *video_transcoder* conversion transcodes the video to the constraints of the terminal (H.264 visual format). Finally, *goal* represents the adapted content.

During the decision process CAIN-21 has determined that the content the terminal accepts may be visual or audiovisual *content*={*visual*, *audiovisual*} (inference rule 1 in Subsection 4.2). This occurs because the *mpeg21:TransportCapabilitiesType* element exists in the terminal description (see Listing 1).

If we change the terminal of this experiment from *iphone* to *http_nokia_n95* we have another didactic experiment. In this case, CAIN-21 produces a sequence with four conversions *initial* → *image_transcoder* → *image_2_video* → *video_transdoder* → *http_delivering* → *goal*. CAIN-21 adds the *http_delivering* conversion to change the *binding* property from *file* to *http*. In the previous experiment the *iphone* terminal supported the *file* delivery mechanism (see Listing 1), which corresponds to the *binding* property at the output of *video_transcoder*. However, the *http_nokia_n95* terminal only supports *binding* ={*http*}. Thus CAIN-21 has added the *http_delivering* conversion that receives a file (i.e., *binding*={*file*} in its preconditions) and delivers it through HTTP (i.e., *binding*={*http*} is its postconditions).

In the second experiment, the decision process needs the *binding_mode* to identify that the *http_delivering* conversion has to be added. This information removes ambiguity and validates extensions 1. In the same way, before adding the *optional* attribute to the *mpeg21:AudioCapabilitiesType* description (extension 2), CAIN-21 did not encounter a sequence for these experiments. This happened because the output of the *image_2_video* conversion did not contain the *audio_format* and *audio_bitrate* properties (see Table 1). Therefore, these properties never matched with the *audio_decoding_format* and *audio_bitrate* properties of the terminal. This lack of matching indicates that the audio constraint is never satisfied. Labelling the audio as *optional* (see Listing 1) allows ignoring the audio properties during the computation of the sequence.

6. Conclusions

Several missing properties from the *mpeg21:TerminalType* that are necessary to make sound decisions during the adaptation of a DI to the *UED* description have been identified. Ambiguity has been demonstrated when this information is absent. The paper has also explained the decision mechanisms that CAIN-21 incorporates and how this mechanism sometimes failed due to ambiguities. Once the *mpeg21:TerminalType* has been extended, the matching process effectively finds sequences of conversions that adapt a particular DI to the terminal.

7. References

[1] I. S. Burnett, F. Pereira, R. V. de Walle, R. Koenen, "The MPEG-21 Book", John Wiley and Sons. 2006.

[2] F. López, J. M. Martínez, N. García. "CAIN-21: An extensible and metadata-driven multimedia adaptation engine in the MPEG-21 framework". LNCS vol. 5887, pp. 114-125 in SAMT09. December 2009.

[3] B. S. Manjunath, P. Salembier, T. Sikora (eds.). "Introduction to MPEG-7 Multimedia Content Description Interface". Wiley & Sons. 2002.

[4] F. López, J. M. Martínez, N. García. "Towards a fully MPEG-21 compliant adaptation engine: complementary description tools and architectural models". LNCS under preparation, Proc. of AMR 2008. June 2008.

[5] D. Jannach, K. Leopold, C. Timmerer, H. Hellwagner, "A Knowledge-Based Framework for Multimedia Adaptation", Applied Intelligence, 24(2):109-125, April 2006.

[6] F. López, D. Jannach, J. M. Martínez, C. Timmerer, H. Hellwagner, and N. García, "Multimedia Adaptation Decisions Modelled as Non-Deterministic Operations", Proc. of WIAMIS'08, pp. 46-49. May 2008.

[7] ISO/IEC 21000-18:2008 FDAM 1"Information technology – Multimedia framework (MPEG-21) – Part- 18: Digital Item Streaming". 2008.