PACE: an Experimental Web-Based Audiovisual Application using FDL

Marc Caillet, Jean Carrive, Vincent Brunie, Cécile Roisin

PACE: an Experimental Web-Based Audiovisual Application using FDL

Marc Caillet, Jean Carrive, Vincent Brunie
Institut National de l'Audiovisuel (INA)
Audiovisual Content Description
4 avenue de l'Europe,
94366 Bry-sur-Marne Cedex, France
{mcaillet,jcarrive,vbrunie}@ina.fr

Cécile Roisin
INRIA Rhône-Alpes & UPMF
WAM Project
655 avenue de l'Europe, Montbonnot,
38334 St-Ismier Cedex, France
{marc.caillet,cecile.roisin}@inrialpes.fr

Abstract

This paper describes the PACE experimental multimedia application that aims at providing automatic tools for web browsing of television program collections; experimentations are currently in progress with a fifty-four "Le Grand Échiquier" show collection. PACE has been built with the FERIA framework and relies on multiple automatic analysis tools. It is generic enough to easily adapt to other collections. Emphasis is made on the new audiovisual documents description language FDL as it is the core part of FERIA, with a particular attention paid on how it operates in PACE.

1. Introduction

Digitization is now coming as a required step in the production or archiving process of audiovisual contents. Moreover, there is a growing number of television programs that are numerically produced. As a result, audiovisual documents are no more analog tapes, they are computer files. Their contents can be thus accessed without knowing neither their physical location nor their physical or logical structure. Queries as well as access to audiovisual content only require descriptions of the content. That is where virtuality lies. Moreover, it is possible to access audiovisual contents from different points of view depending of what is sought for. With such new access modes, new multimedia applications of a great interest for audiovisual resources holders are rising.

This paper focuses on the PACE[1] experimental application that aims at automatically publishing television program collections on the web. Section 2 describes its objectives and functionalities as well as the FERIA[2] project which PACE has been developed with. For virtual audiovisual contents to be accessed through descriptions, a description language is needed. Section 3 briefly presents a new au-

diovisual document description language, FDL[3], that forms the core part of the FERIA framework. Section 4 describes how it operates in PACE and what PACE is gaining from it. Conclusion and future developments are the subject of Section 5.

2. The PACE application

INA[4] has been archiving and indexing broadcasted French television and radio programs for thirty years and has thus to manage huge audiovisual resources. According to the requests it is daily receiving from individuals, the most promising market for television and radio programs archives lies in excerpts. For this market to be profitable, low cost and generic audiovisual contents browsing applications must be provided. That is exactly what the experimental PACE application is aiming at.

PACE provides a generic way to automatically publish television program collections on the web. It also allows individuals to navigate by queries through these collections and thus to easily discover excerpts. Experimentations are currently in progress with a fifty-four "Le Grand Échiquier"[5] show collection. Anyone who is seeking for an excerpt of one of these shows through PACE is provided with many browsing methods: he is first able to consider either the whole collection or a specific show; he is then able to ask for the faces of the guests that appear in the collection or in the specific show he previously selected, or ask for the faces of guests that may have said words of his interest, or even for broached topics; he is finally able to ask for sequences in which a specific guest appears either as a performer or as an interviewee, or sequences about a specific topic of his interest. And the process may go on and on as our excerpts seeker is now able, at this stage of his research, to ask for the faces of guests that appear in — or the broached topics of — the returned sequences and so on.

---

[1] Automatic Publishing of a Television Program Collection.
[2] Framework for experimentation and industrial production of multimedia applications.

[3] Feria Description Language.
[4] French National Institute of Audiovisual.
[5] One of the most famous French variety show (from 01/72 to 12/89).

It is important to note that the queries are hard-wired so that the excerpts seeker does not need to *a priori* know what he is seeking for. This *parti pris* allows him to discover the audiovisual contents of the collection by strolling through it, without being required to know any query language.

PACE is being developed within the FERIA framework whose purpose is to design low cost multimedia applications based on automatic analysis (some industrial applications, such as MANREO or KINOMAI[6], are partly sharing some features with FERIA). Figure 1 depicts its global architecture.
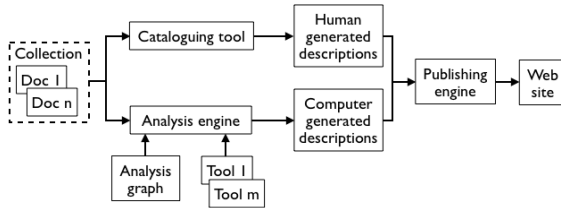

Figure 1. PACE architecture.

INA archivists annotate the documents with cataloguing information, *e.g.* the director's name. The analysis engine runs the automatic analysis tools — video and audio analysis (*e.g.* [2]), and audio classification (*e.g.* [5]) — according to an analysis graph (see figure 2). It produces many different viewpoints on a given television program, and thus as many descriptions. Both human- and computer-generated descriptions are processed by the publishing engine to automatically produce a web site. This architecture makes PACE generic enough to easily adapt to other television program collections.

3. FDL, a new audiovisual documents description language

FDL forms the core part of the FERIA framework as its audiovisual document description language. How came we proposed a new description language whereas describing audiovisual documents is what the *Multimedia Content Description Interface* ISO standard, known as MPEG-7 ([3] and [4]), is designed for? While this question is extensively discussed in a previous work of our own ([1]), the following draws a general survey of it.

3.1   FDL, an alternative to MPEG-7

Requirements for an audiovisual description language are resulting from the specific needs of multimedia applications such as PACE:

- expressiveness power: taxonomic hierarchy of descriptors as a way to express their semantics; ability to create a new descriptor by structuring other descriptors; link to the media independently of its physical location;

---
[6]see respectively www.netia.net/index.php and www.kinomai.com

- descriptor processing: specification of descriptors as description classes that can instantiated; controlled extensibility and modularity;

- platform- and application-independent syntax.

Because our own requirements are close to those of MPEG-7, we first evaluated this standard as an applicant to fulfill them and came to the conclusion that it doesn't meet them all. One of the major drawback of MPEG-7 lies in its inability to taxonomically hierarchize the descriptors and thus its inability to express their semantics. This drawback is due to the choice of XML Schema as MPEG-7 DDL (Definition Description Language). Other drawbacks are MPEG-7 non-modularity and non-extensibility — or rather MPEG-7 inability to validate extensions of its description schemes. These drawbacks harshly shake MPEG-7 deepest foundations; the problem is too important to be solved by simply modifying the standard. We thus opted for a paradigm switch: we switched from a documentary one in which the description language allows the syntactic expression of audiovisual document annotations (*e.g.* MPEG-7) to a knowledge representation one in which audiovisual descriptors are objects organized in a taxonomic hierarchy that holds semantics.

3.2   FDL, a short description

FDL is an object language. It can be thought about as a meta-language that allows the definition of application-specific audiovisual document description languages. All of these FDL-languages share a few constants:

- Only three semantically related generic descriptors on the basis of which other descriptors may be built: `Descriptor` is the root descriptor which any other descriptors inherits from; `TemporalDescriptor` inherits properties from `Descriptor` and adds a temporal location property; `SpatialDescriptor` inherits properties from `TemporalDescriptor` and adds a spatial location property.

- Extensibility mechanisms: inheritance of properties and composition of descriptors. These mechanisms are fully under FDL control so that any user-defined descriptor can be validated.

Such a short number of descriptors truly contrasts with MPEG-7 six hundred multimedia description schemes. Together with the extensibility mechanisms, they provide FDL with a major asset: they allow to freely design consistent descriptors that fit any application specific needs, thus making FDL readily adaptable to new multimedia applications to come. Using MPEG-7, instead, would have led to either try to adapt to predefined descriptors or to design extensions of them that could not be validated.

Other FDL constants are: link to the media, spatial and temporal localisation types, basic types, matrix of any di-

mension type. Altogether, they allow to define: descriptors as description classes (DCs) and descriptions (Ds) as instances of description classes.

XML is the syntax for FDL. It has been opted for in order to facilitate the storage and the management of description classes and descriptions, and to allow the use of well-established standard XML tools to perform some validation tasks.

4. FDL for PACE

FDL is used in FERIA to syntactically and semantically express the descriptors. Figure 2 depicts the analysis graph that leads to PACE. The directed edges outline the FDL descriptions flow: some FDL descriptions are used as inputs to analysis tools, some others are directly used by PACE.
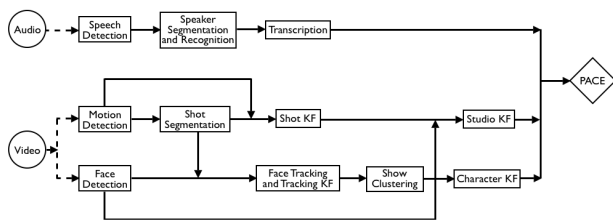

Figure 2. PACE analysis graph (KF stands for Key Frame).

This section focuses on the results of the shot segmentation tool. It describes how the `ShotSegmentationToolResult` descriptor and one of its instance are coded in FDL, and how they are validated. It ends with a discussion about what PACE is gaining from FDL.

4.1 The ShotSegmentationToolResult description class and an instance

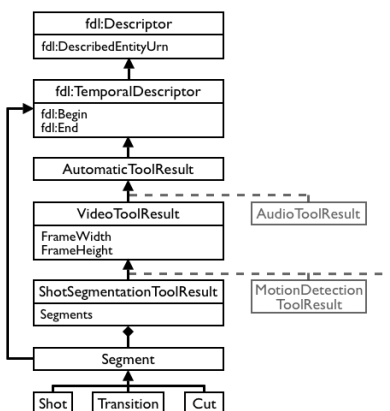Figure 3 shows the UML modelling schema of the `ShotSegmentationToolResult` description class.

Figure 3. UML schema of ShotSegmentationToolResult.

The `AutomaticToolResult` is an abstract description class, *i.e.* it cannot be instantiated, that subsumes the audio and video description classes that are used for PACE. The script below shows a sample of its FDL code:

```
<DescriptionClass ... >
 <DescriptionClassHeader>
  <Urn>urn:x-feria:dc:automatictoolresult</Urn>
 </DescriptionClassHeader>
 <DescriptionClassBody>
  <Descriptor id="x-0">
   <Name>AutomaticToolResult</Name>
   <Parent>fdl:TemporalDescriptor</Parent>
   <DescriptorAttributes>Abstract</DescriptorAttributes>
  </Descriptor>
 </DescriptionClassBody>
</DescriptionClass>
```

An FDL description class is divided into two distinct parts: the header specifies the urn of the descriptor; the body defines the descriptor itself which carries an `id` attribute (its id is `x-0` in the example above) so that it can be referred to by other descriptors.

Inheriting. The `Parent` element creates an inheritance relationship between the `AutomaticToolResult` descriptor and the `fdl:TemporalDescriptor` FDL generic descriptor. Note that every user-defined descriptor must inherit from one of the three FDL generic ones to be deemed to be a new FDL descriptor.

The `VideoToolResult` description class subsumes all video description classes PACE makes use of. The first FDL sample code below shows how the `VideoToolResult` descriptor inherits from the `AutomaticToolResult` descriptor:

```
<DescriptionClass ...
xmlns:atr="urn:x-feria:dc:automatictoolresult" ... >
 ... <Parent>atr:x-0</Parent> ...
```

As for the previous descriptor, the inheritance link is created by the `Parent` element. In the sample code above, it is valued to `atr:x-0`: on the left side, `atr` is a shortcut that refers to the `AutomaticToolResult` description class through its urn; on the right side, `x-0` refers to the id of the descriptor which `VideoToolResult` inherits from.

Adding a new property. The second FDL sample code of the `VideoToolResult` description class shows how the `FrameWidth` property is added:

```
<Property id="p-0">
 <Name>FrameWidth</Name>
 <Range type="integer"/>
</Property>
```

Structuring. The following FDL sample code of the `ShotSegmentationToolResult` description class emphasizes how structures are being used:

```
<DescriptionClass ...
 xmlns:sgmt="urn:x-feria:dc:segment" ... >
 ...
  <Structure id="s-0" use="required">
   <Name>Segments</Name>
   <Type>fdl:TemporalStructure</Type>
   <Descriptor id="d-1" import="sgmt:d-0"/>
  </Structure>
...
```

The `Type` element points out that the `Segments` structure is of type `fdl:TemporalStructure`, *i.e.* that it is a list of temporal descriptors without any other constraint. The `Descriptor` element that follows forces upon the

structure it is enclosed within to only accept temporal descriptors of a specific type. The `import` attribute valued to `sgmt:d-0` specifies that these descriptors must be `Segment` ones, and that the `Segment` description class (that subsumes `Shot`, `Transition` and `Cut` as depicted in figure 3; its FDL code is not given in this paper) is located elsewhere.

Instancing. Finally, the sample code below is an instance of the `ShotSegmentationToolResult` descriptor where all the properties and structure are instantiated :

```
<fdl:Description ...>
 ...
 <fdl:DescriptionBody>
  <ShotSegmentationToolResult id="shotsegment-0001">
   <fdl:DescribedEntityUrn>
    urn:x-feria:doc:ina:CPB81050169
   </fdl:DescribedEntityUrn>
   <fdl:Begin>00:00:00:00000000</fdl:Begin>
   <fdl:End>00:00:30:00000000</fdl:End>
   <FrameWidth>352</FrameWidth>
   <FrameHeight>288</FrameHeight>
   <Segments id="segments">
    <s:Shot id="shot-1">
     <fdl:Begin>00:00:00:00000000</fdl:Begin>
     <fdl:End>00:00:02:00000000</fdl:End>
    </s:Shot>
    <t:Transition id="trans-1">
     <fdl:Begin>00:00:02:00000000</fdl:Begin>
     <fdl:End>00:00:03:00000000</fdl:End>
    </t:Transition>
    ...
```

4.2 Validation

FDL has been implemented within the .NET framework. Briefly, a description class parser dynamically compiles description classes to dynamic libraries, and a description parser loads the relevant classes and dynamically instantiates them. Both description classes and descriptions are validated while being parsed, as figure 4 illustrates it for the `ShotSegmentationToolResult` (SSTR).
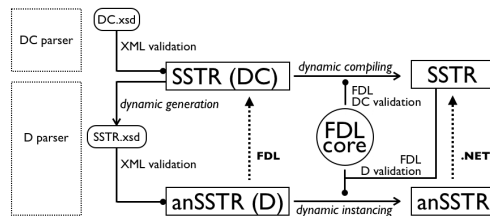

Figure 4. The validation process.

XML schema is used to syntactically constraint the description classes, and to validate a part of the logical structure and data types of the descriptions. FDL completes the validation process by taking charge of what XML schema is unable to express. For example: checking that the descriptor is a direct or undirect descendant of one of the three FDL generic descriptors, checking that the structures constraints are fulfilled. The `SSTR.xsd` schema is automatically generated while the `SSTR` description class is parsed.

4.3 What is PACE gaining from FDL?

At a given step of his investigation, the excerpts seeker we considered in section 1 is provided with the ability to ask either for sequences of any type, or for performance or interview ones. Because the FDL underlying object model allows to model both performance and interview sequences as sub-classes of the sequence class (so that instances of these two classes are also instances of the sequence class; this semantic link cannot be expressed using MPEG-7), it is easy for PACE to answer to the corresponding queries. FDL is also able to validate extensions of its predefined descriptors, whereas MPEG-7 is not. These are two great features because they allow us to design our own consistent descriptors. These descriptors may have been designed using MPEG-7 but without semantic links between them and without the ability to be validated.

Moreover, FDL is fully modular. So that both tools and FDL descriptors are reuseable in FERIA to design other multimedia applications. For example, `FaceDetection`, `StudioKF`, `SpeechDetection` and `Speaker Segmentation and Recognition` are also part of the automatic analysis tools sequence that leads to the FIDELIO/ALTO[7] experimental applications.

5. Conclusion and future works

This paper showed how PACE has been designed with the FERIA framework and what it is gaining from FDL. Every analysis tool works correctly and produces descriptions in FDL. The analysis graph has been "manually" validated. However, the PACE application is still work in progress. Integration of the tools within the FERIA framework is left to be done. Moreover, FDL itself is evolving to enhance the definition and control of structures. For example, in PACE, more constrained structures would be of a great usefulness for the `Segments` structure to constrain its descriptors not to time-overlap. Dealing with such constraints is going to be an important part of our short term future work.

References

[1] M. Caillet, J. Carrive, and C. Roisin. Description des documents audiovisuels : s'affranchir des limitations de mpeg-7. In *Proceedings of MetSI05*, pages 69–78, May 2005.

[2] B. Fauvet, P. Bouthemy, and P. Gros. A geometrical keyframe selection method exploiting dominant motion estimation in video. In *Proceeding of CIVR*, pages 419–427, 2004.

[3] J. M. Martínez, R. Koenen, and F. Pereira. Mpeg-7: The generic multimedia content description standard, part 1. *IEEE Multimedia*, 9(2):78–87, April–June 2002.

[4] J. M. Martínez, R. Koenen, and F. Pereira. Mpeg-7: The generic multimedia content description standard, part 2. *IEEE Multimedia*, 9(3):83–93, July–September 2002.

[5] J. Pinquier and R. André-Obrecht. Audio classification tools for multimedia indexing. In *Proceedings of CBMI*, 2005.

[7]FIDELIO enhances an opera show with metadata, ALTO provides a way to use thse metadata in an interactive television program broadcast.