

# **PACE: an Experimental Web-based Audiovisual Application using FDL**

Marc Caillet (mcaillet@ina.fr)

INA, DCA team

INRIA, WAM project

WDA 2005, Seoul, South Korea



## **Talk outline:**

- 1. Context and Motivation**
- 2. The PACE application**
- 3. FDL, a new audiovisual description language**
- 4. FDL for PACE**
- 5. What is PACE gaining from FDL?**
- 6. Conclusion and future work**

## 1. Context and Motivation

### **INA - French National Institute of Audiovisual:**

- ➔ Archiving and indexing television and radio programs.
- ➔ Commercializing to broadcast professionals and general public (a program collection, a specific program, an excerpt).
- ➔ Digitizing (since 2001):
  - Incoming stream is digitally recorded.
  - Archives are being stepwise digitized.

### **Digitized audiovisual contents:**

- ➔ Audiovisual contents are now virtual.
- ➔ Contents can be accessed through descriptions.

## 1. Context and Motivation

### INA specific needs:

- ➔ Management of huge audiovisual document collections.
- ➔ Bring the whole audiovisual document collection to the attention of potential customers.

### Digitization:

- ➔ Considering efficient answers to these needs with the use of automatic analysis tools:
  - signal analysis
  - symbolic computation on descriptions produced by other tools

### What is new?

- ➔ Huge digital audiovisual collection (over 500,000 hours).
- ➔ Satisfying these needs is now a urgent necessity.

## **2. The PACE application**

### **Purpose of PACE:**

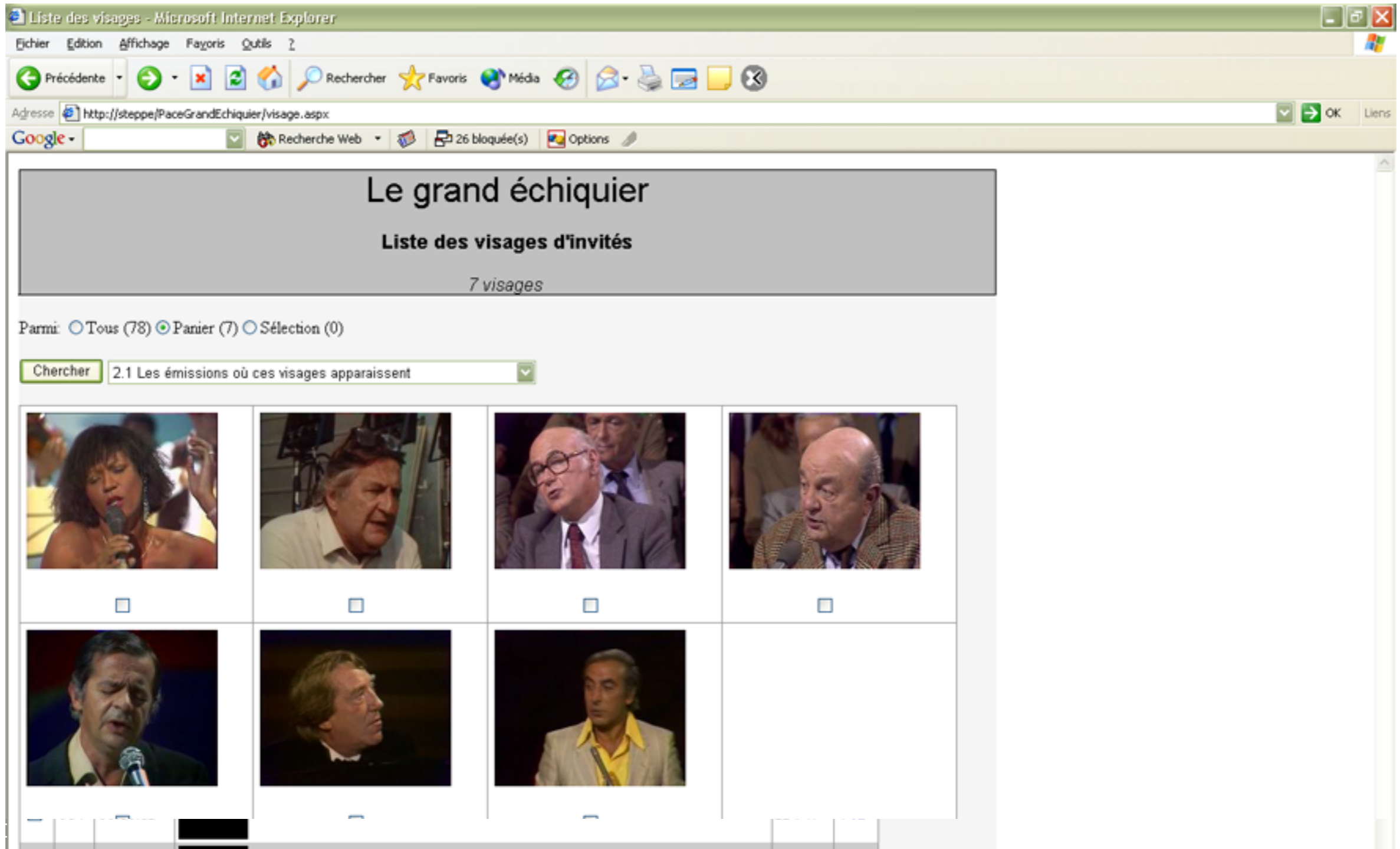
- ➔ Bring audiovisual program collections to the attention of potential customers.
- ➔ Make potential customers able to retrieve excerpts of interest.

### **Status of PACE:**

- ➔ Experimental generic web-based application.
- ➔ Experimentations with a 54 “Le Grand Échiquier” program collection.
- ➔ PACE is still work in progress.

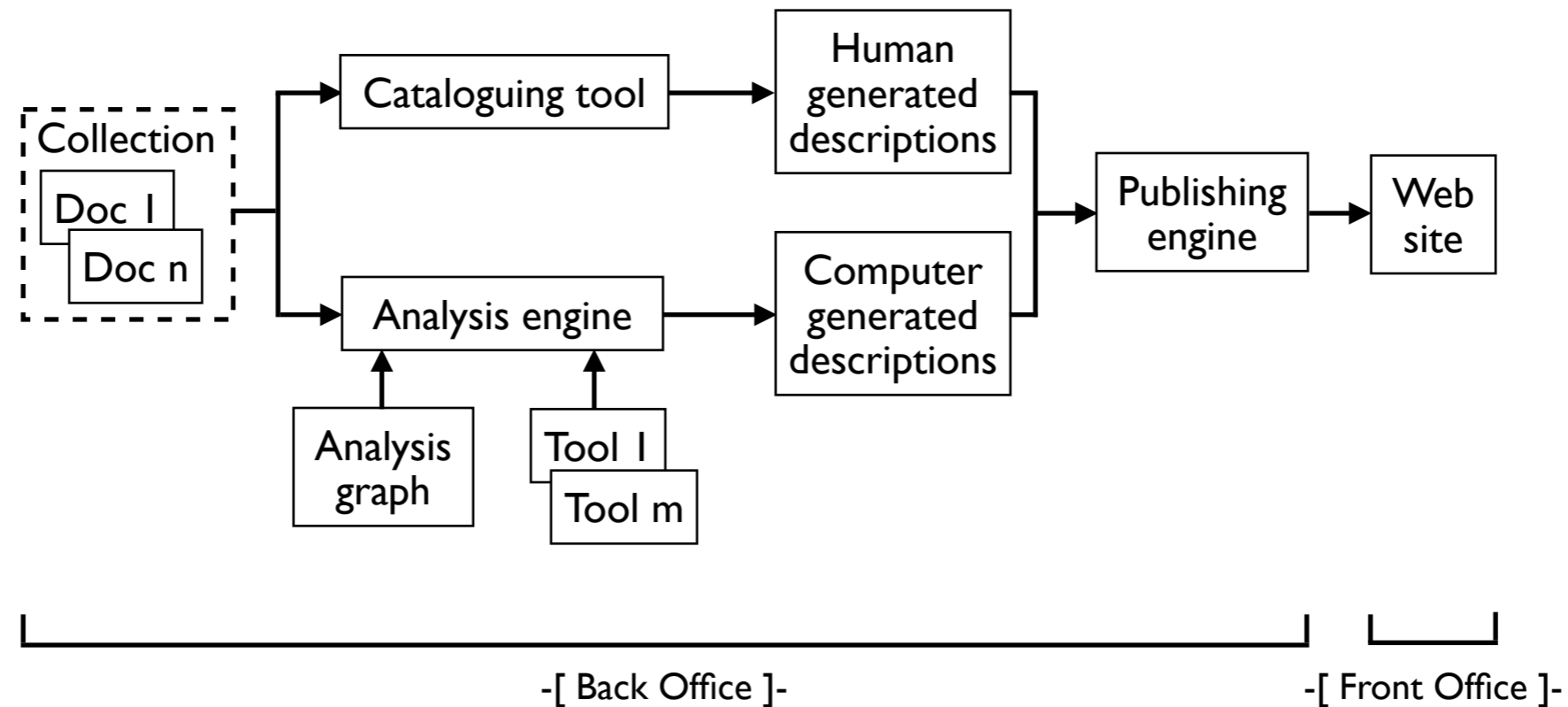
## 2. The PACE application

### An Overview of PACE (a prototype)



## 2. The PACE application

### Architecture of PACE:



**Cataloguing and analysis tools both produce descriptions of audiovisual documents in FDL**

### 3. FDL, a new audiovisual description language

#### Requirements for an audiovisual description language:

- ➔ Expressiveness power:
  - Semantics of descriptors (taxonomical hierarchy)
  - Structuring descriptors
  - Link to the media
  
- ➔ Descriptor processing:
  - Description classes and descriptions
  - Extensibility
  - Modularity
  
- ➔ Application and platform independence.

**These requirements are close to those of MPEG-7**



### 3. FDL, a new audiovisual description language

#### MPEG-7, Multimedia Content Description Interface:

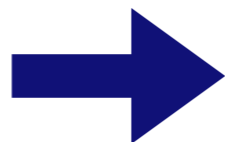
- ➔ Standardizes a set of description tools:
  - *Descriptors* (Ds): describing low level audiovisual features
  - *Description Schemes* (DSs): describing complex multimedia entities
  - *Description Definition Language* (DDL): coding Ds and DSs
- ➔ MPEG-7 drawbacks:
  - Inability to taxonomically hierarchize descriptors (no semantics)
  - Inability to validate extensions
  - Non-modularity

**MPEG-7 does not meet the requirements  
for an audiovisual description language**

#### MPEG-7

Annotations  
Metadata

**Documentary**



#### FDL

Knowledge Representation  
Object Modelling

**Artificial Intelligence**

### 3. FDL, a new audiovisual description language

#### FDL: object modelling

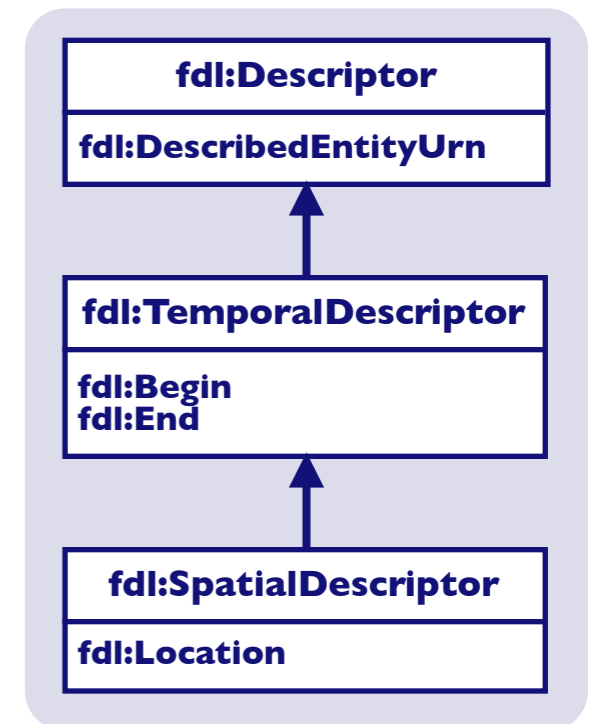
➔ FDL is an object language

➔ FDL core :

- Only three semantically linked generic descriptors
- Structures
- Extension mechanisms
- ...

➔ These features allow to express,  
with an XML syntax:

- Description Classes (DCs): a description class defines a descriptor.
- Descriptions (Ds): a description is an instance of a descriptor.



### 3. FDL, a new audiovisual description language

## FDL: meeting the requirements for an audiovisual description language

➔ 2 extension mechanisms under FDL control:

- Inheritance of properties
- Structuring

**Extensibility**

➔ Inheritance of properties (taxonomic hierarchy) and structuring allow to express descriptor semantics.

**Semantics**

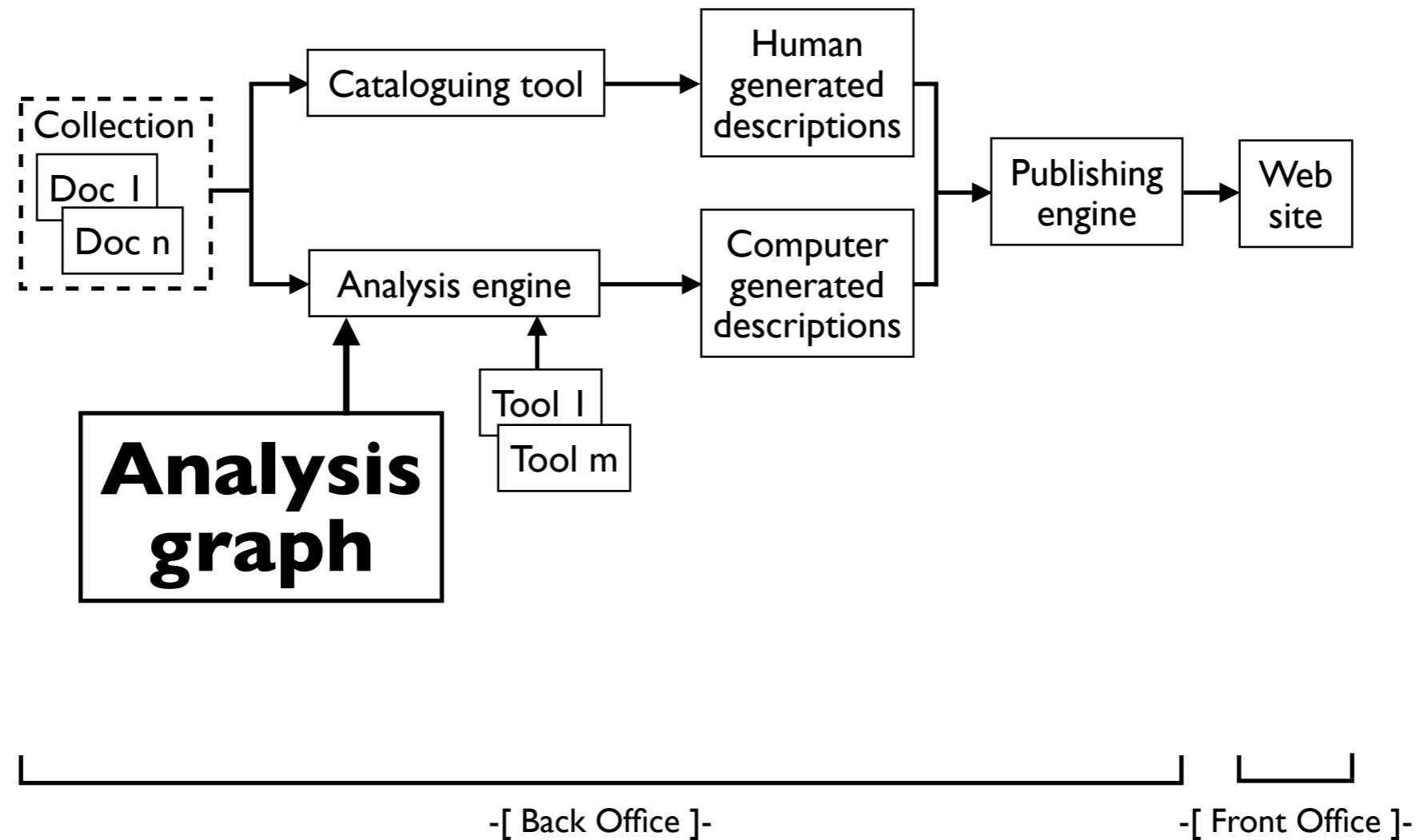
➔ Modularity results from FDL extension mechanisms.

**Modularity**

**FDL meets the requirements for an audiovisual description language**

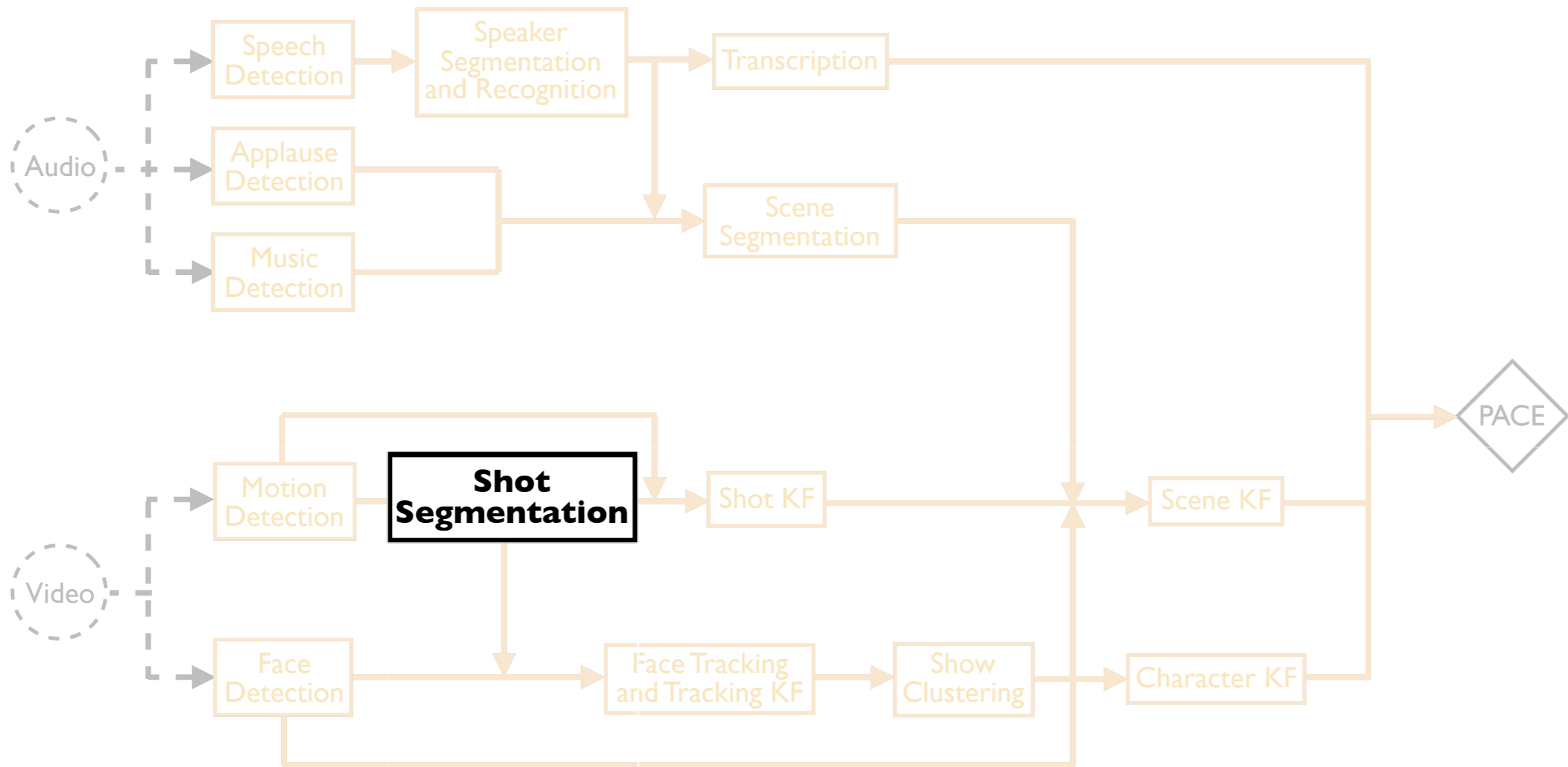
## 4. FDL for PACE

### Architecture of PACE:



## 4. FDL for PACE

### Analysis graph of PACE:

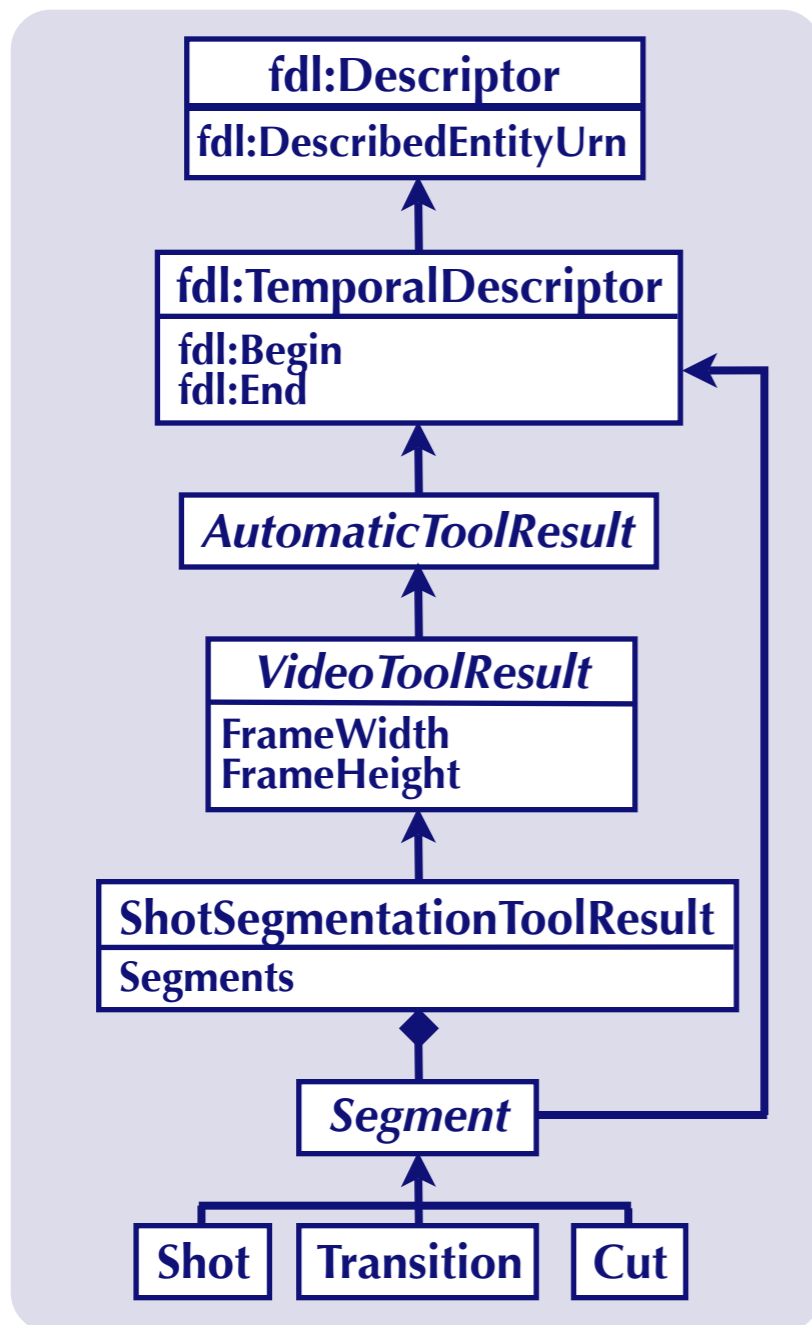


**FDL descriptions are either used by other analysis tools  
or directly by PACE**

## 4. FDL for PACE

### Example of an FDL descriptor:

➔ ShotSegmentationToolResult: object modelling and DC

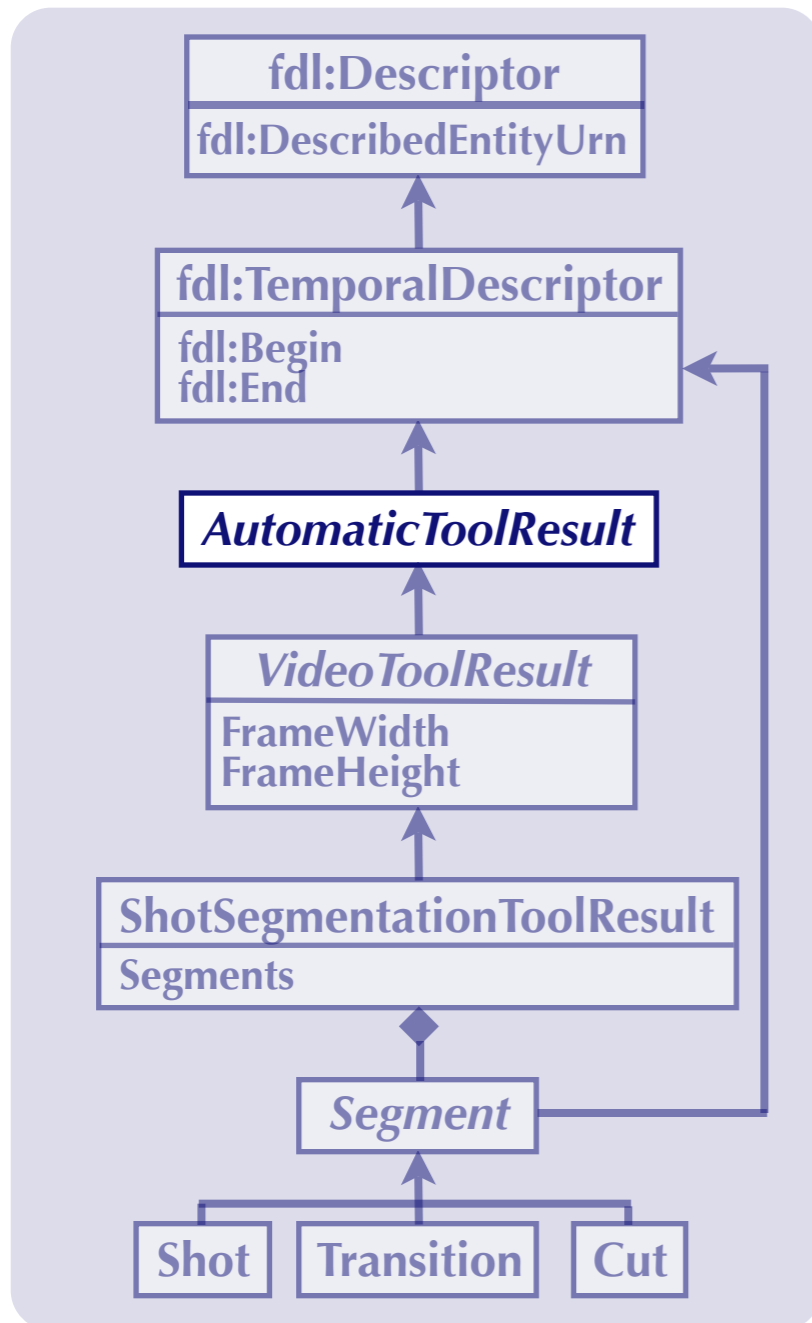


## 4. FDL for PACE

### Example of an FDL descriptor:

➔ ShotSegmentationToolResult: object modelling and DC

automaticTR\_DS.xml



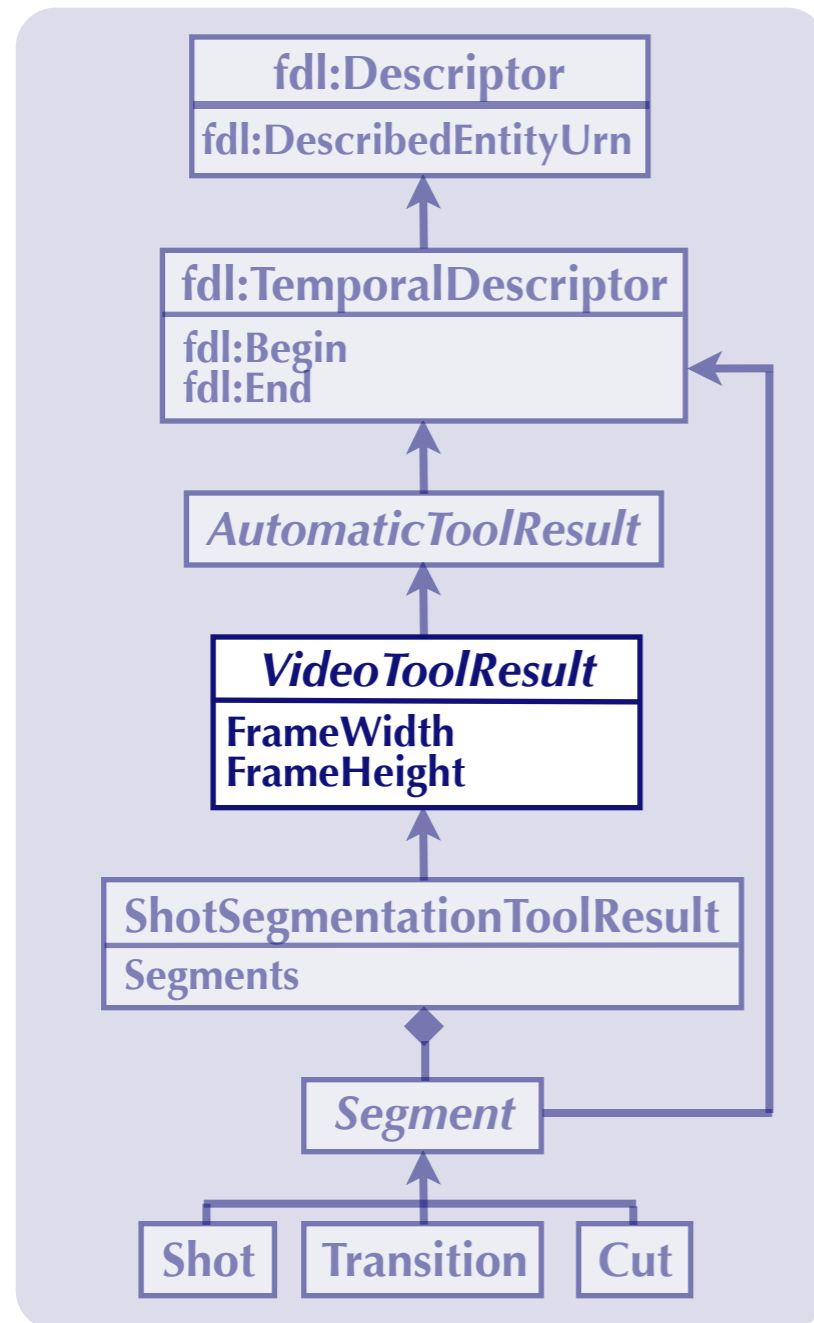
```
<DescriptionClass
targetNamespace="urn:x-feria:dc:automatictoolresult"
xmlns="urn:x-feria:fdl"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:x-feria:fdl fdl.xsd">
  <DescriptionClassHeader>
    <Urn>urn:x-feria:dc:automatictoolresult</Urn>
  </DescriptionClassHeader>
  <DescriptionClassBody>
    <Descriptor id="x-0">
      <Name>AutomaticToolResult</Name>
      <Parent>fdl:TemporalDescriptor</Parent>
      <DescriptorAttributes>Abstract</DescriptorAttributes>
    </Descriptor>
  </DescriptionClassBody>
</DescriptionClass>
```

## 4. FDL for PACE

### Example of an FDL descriptor:

➔ ShotSegmentationToolResult: object modelling and DC

videoTR\_DS.xml



```

<DescriptionClass
  targetNamespace="urn:x-feria:dc:videotoolresult"
  xmlns="urn:x-feria:fdl"
  xmlns:atr="urn:x-feria:dc:automatictoolresult"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:x-feria:fdl fdl.xsd">
  <DescriptionClassHeader>
    <Urn>urn:x-feria:dc:videotoolresult</Urn>
    <ImportDC urn="urn:x-feria:dc:automatictoolresult"/>
  </DescriptionClassHeader>
  <DescriptionClassBody>
    <Descriptor id="y-0">
      <Name>VideoToolResult</Name>
      <Parent>atr:x-0</Parent>
      <DescriptorAttributes>Abstract</DescriptorAttributes>
      <Property id="p-0">
        <Name>FrameWidth</Name>
        <Range type="integer"/>
      </Property>
      <Property id="p-1">
        <Name>FrameHeight</Name>
        <Range type="integer"/>
      </Property>
    </Descriptor>
  </DescriptionClassBody>
</DescriptionClass>
  
```

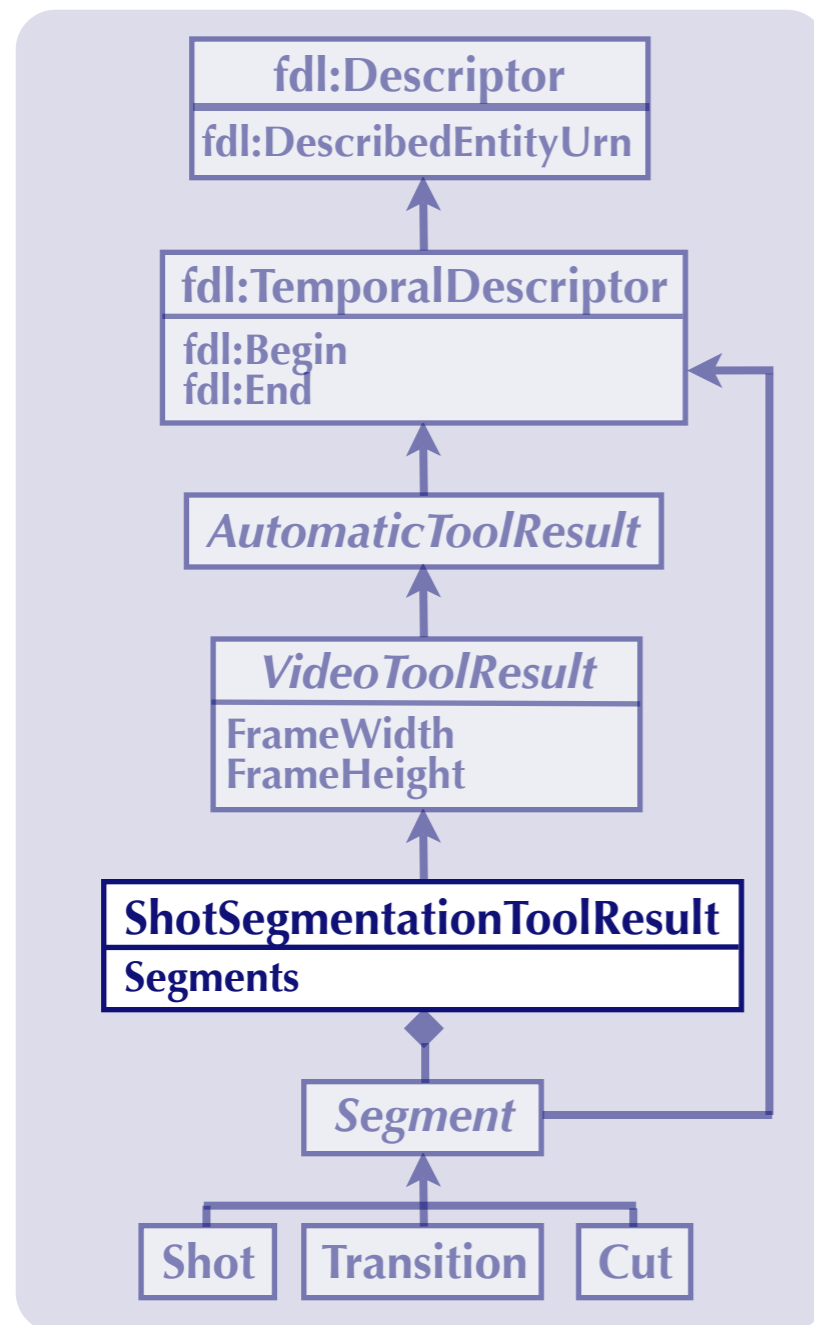


## 4. FDL for PACE

### Example of an FDL descriptor:

➔ ShotSegmentationToolResult: object modelling and DC

shotSegmentationTR\_DS.xml



```

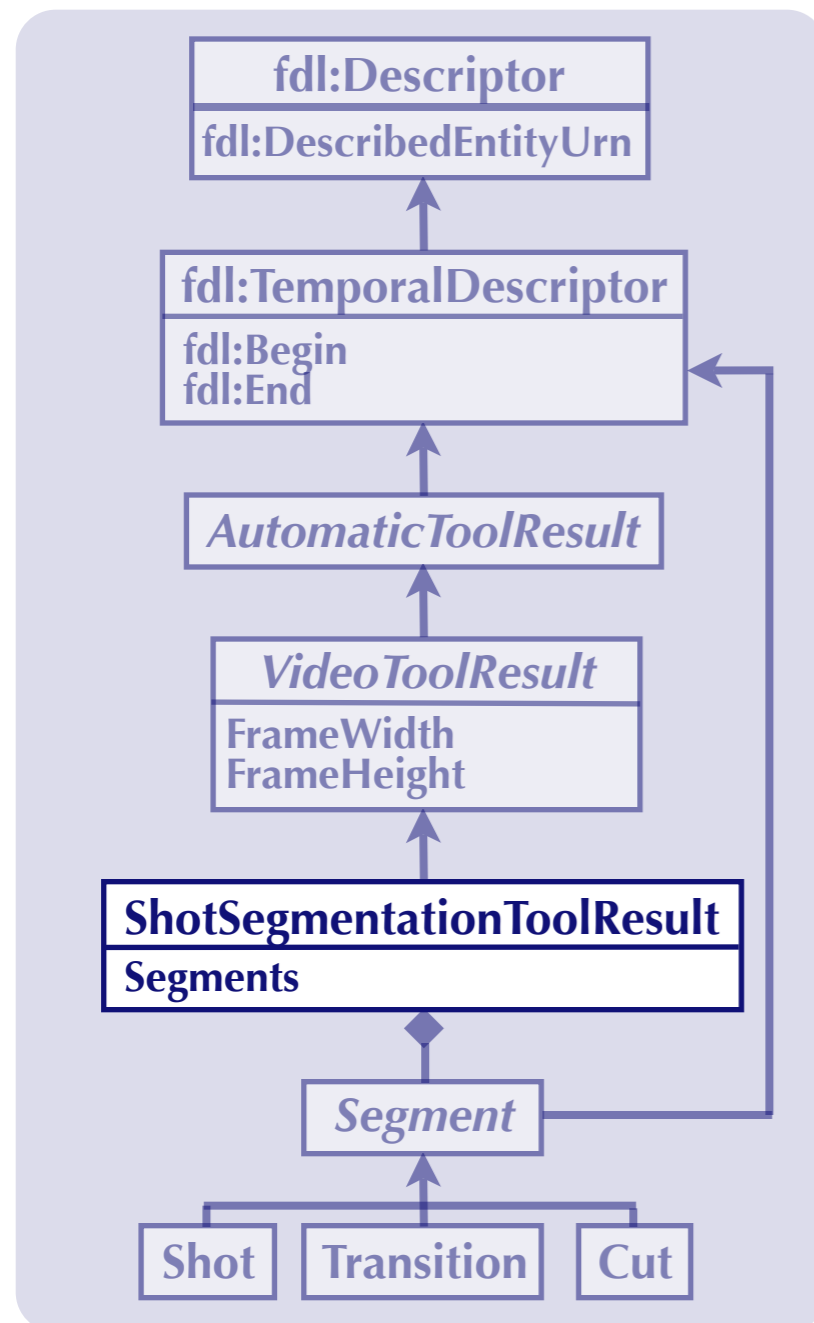
<DescriptionClass
  targetNamespace="urn:x-feria:dc:shotsegmentation"
  xmlns="urn:x-feria:fdl"
  xmlns:video="urn:x-feria:dc:videotoolresult"
  xmlns:sgmt="urn:x-feria:dc:segment"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:x-feria:fdl fdl.xsd">
  <DescriptionClassHeader>
    <Urn>urn:x-feria:dc:shotsegmentation</Urn>
    <ImportDC urn="urn:x-feria:dc:videotoolresult"/>
    <ImportDC urn="urn:x-feria:dc:segment"/>
  </DescriptionClassHeader>
  <DescriptionClassBody>
    <Descriptor id="z-0">
      <Name>ShotSegmentationToolResult</Name>
      <Parent>video:y-0</Parent>
      <Structure id="s-0" use="required">
        <Name>Segments</Name>
        <Type>fdl:TemporalStructure</Type>
        <Descriptor id="d-1" import="sgmt:d-0"/>
      </Structure>
    </Descriptor>
  </DescriptionClassBody>
</DescriptionClass>
  
```

## 4. FDL for PACE

### Example of an FDL descriptor:

➔ ShotSegmentationToolResult: an instance (D)

shotSegmentation\_D0001.xml



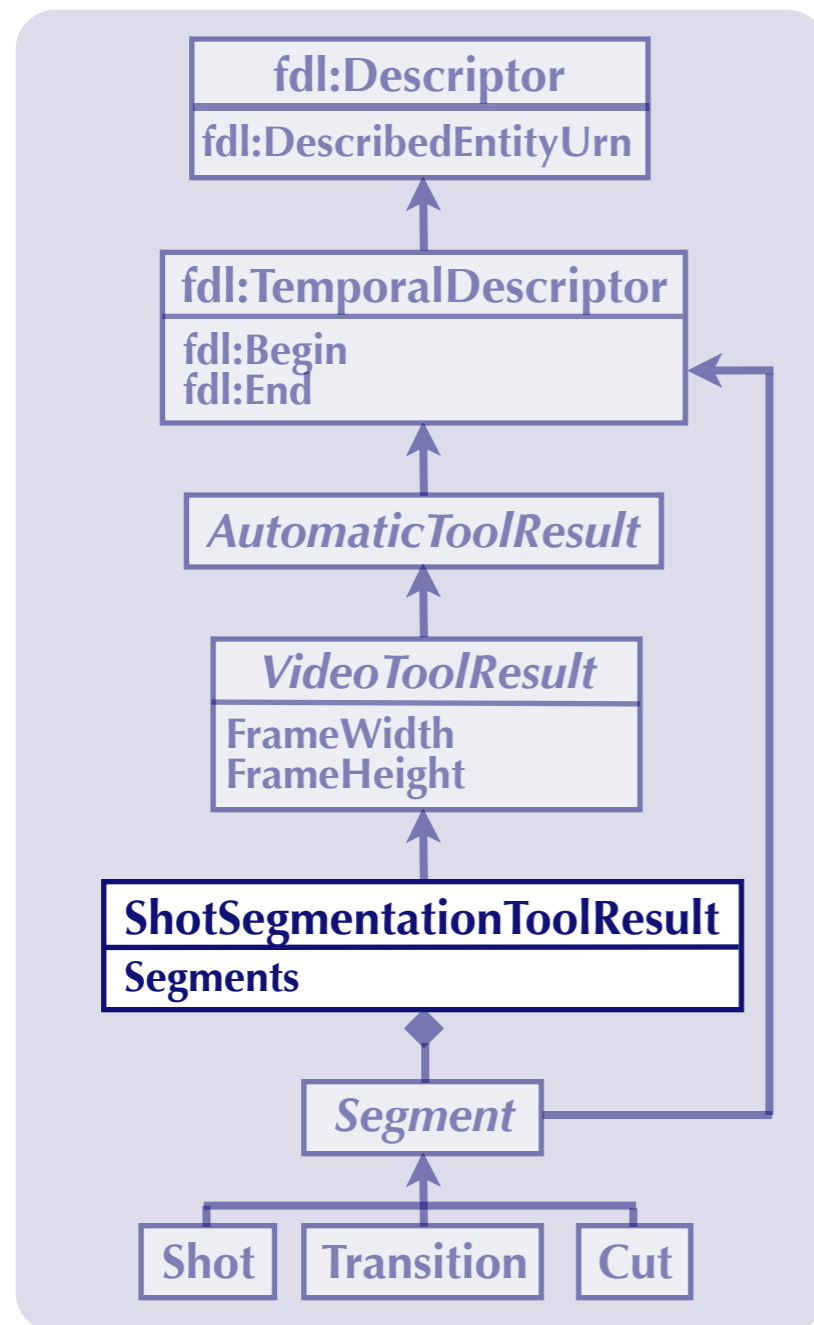
```
<fdl:Description
xmlns="urn:x-feria:dc:shotsegmentation"
xmlns:fdl="urn:x-feria:fdl"
xmlns:s="urn:x-feria:dc:shot"
xmlns:t="urn:x-feria:dc:transition"
xmlns:c="urn:x-feria:dc:cut"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:x-feria:dc:shotsegmentation
shotSegmentation.xsd">
  <fdl:DescriptionHeader>
    ...
  </fdl:DescriptionHeader>
  <fdl:DescriptionBody>
    ...
  </fdl:DescriptionBody>
</fdl:Description>
```

## 4. FDL for PACE

### Example of an FDL descriptor:

➔ ShotSegmentationToolResult: an instance (D)

shotSegmentation\_D0001.xml

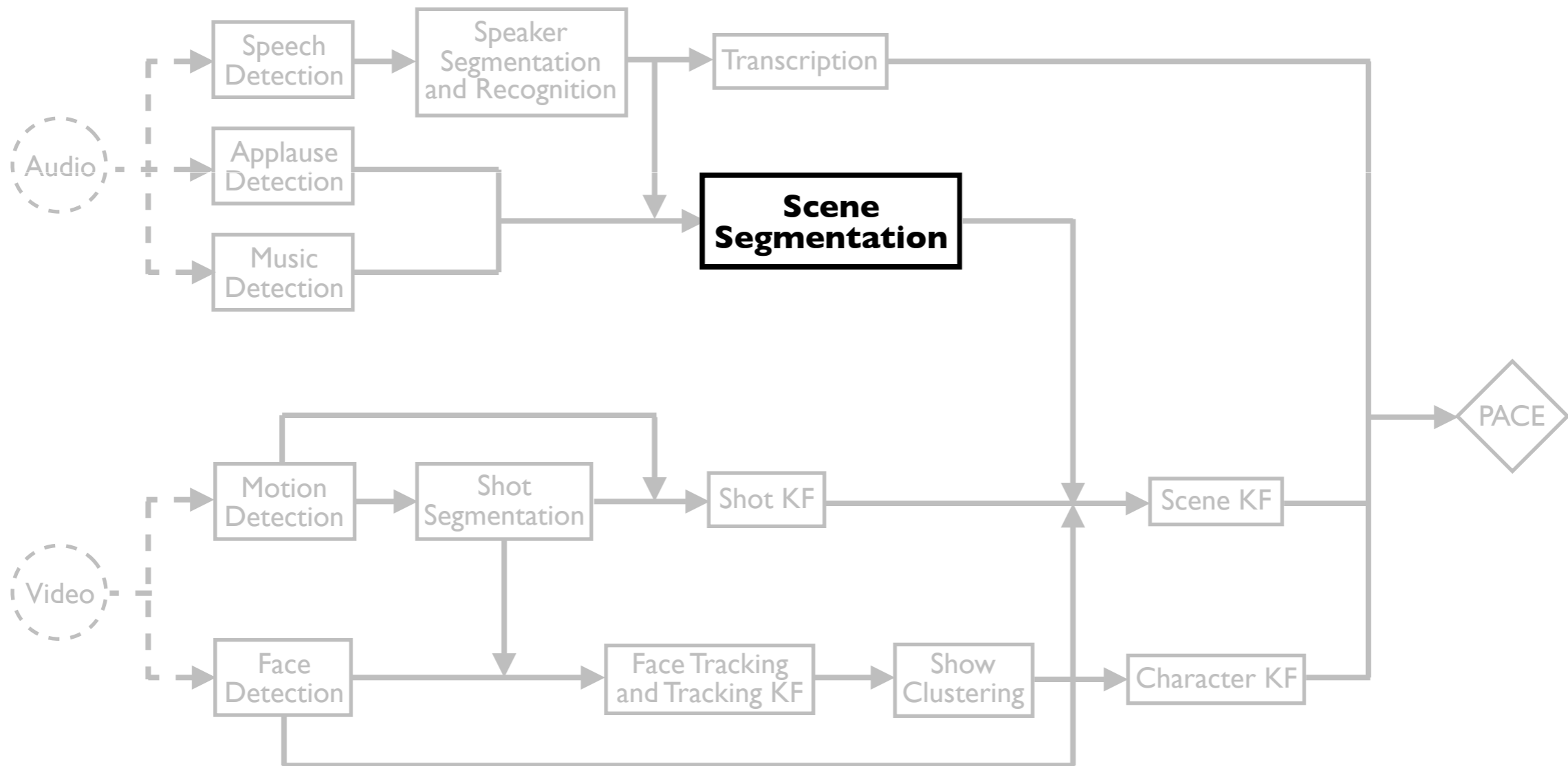


```

<fdl:Description ...>
  <fdl:DescriptionHeader>
    ...
  </fdl:DescriptionHeader>
  <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>00:00:07:00000000</fdl:End>
        </s:Shot>
        <t:Transition id="trans-1">
          <fdl:Begin>00:00:07:00000000</fdl:Begin>
          <fdl:End>00:00:08:00000000</fdl:End>
        </t:Transition>
        ...
      </Segments>
    </ShotSegmentationToolResult>
  </fdl:DescriptionBody>
</fdl:Description>
  
```

## 4. FDL for PACE

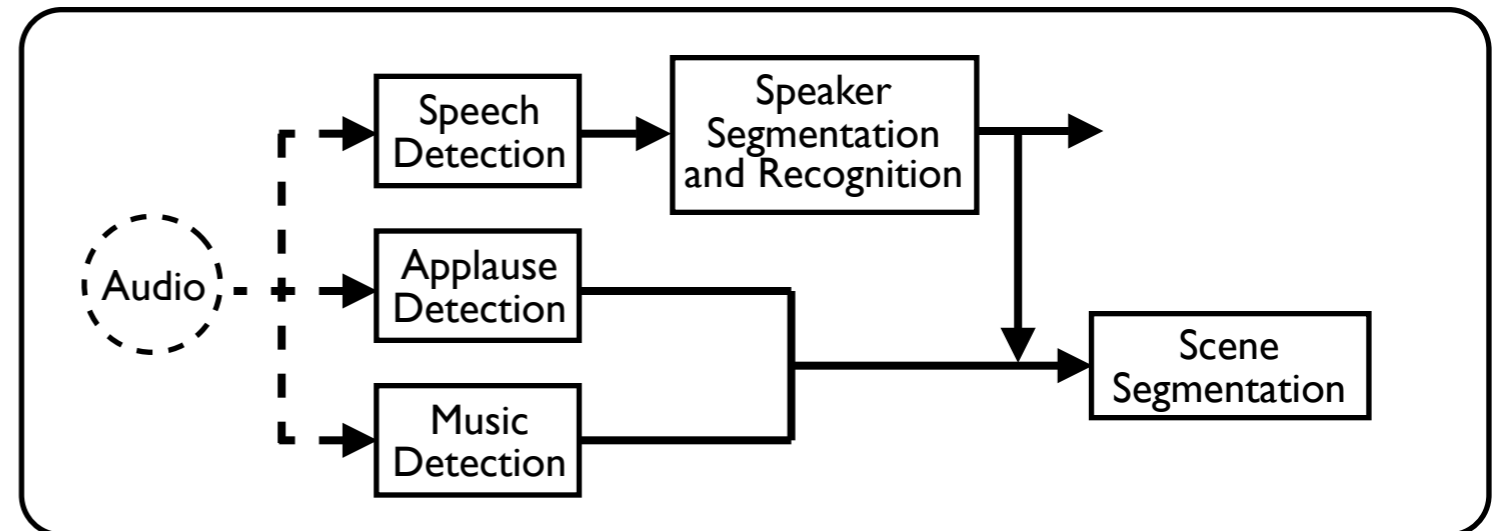
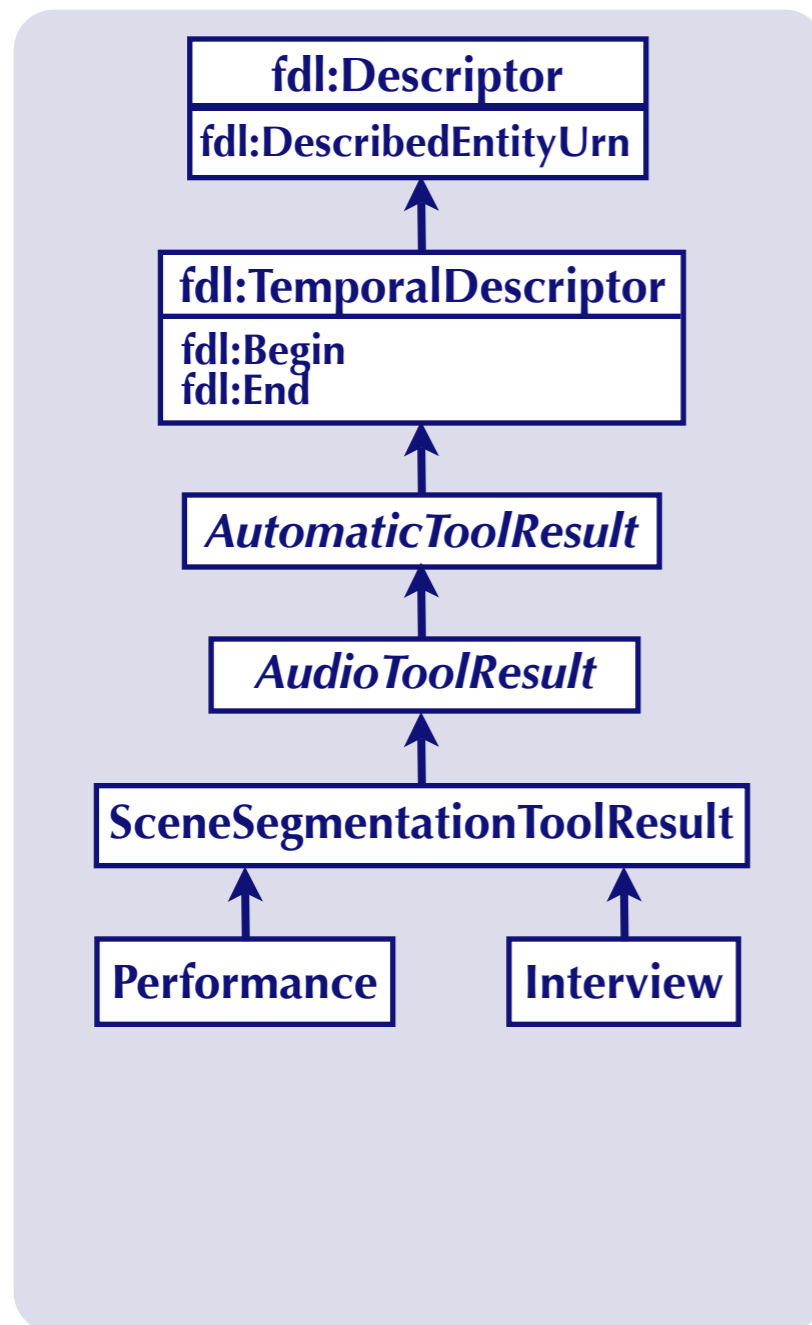
### Analysis graph of PACE:



## 4. FDL for PACE

### A more abstract FDL descriptor:

➔ SceneSegmentationToolResult: object modelling



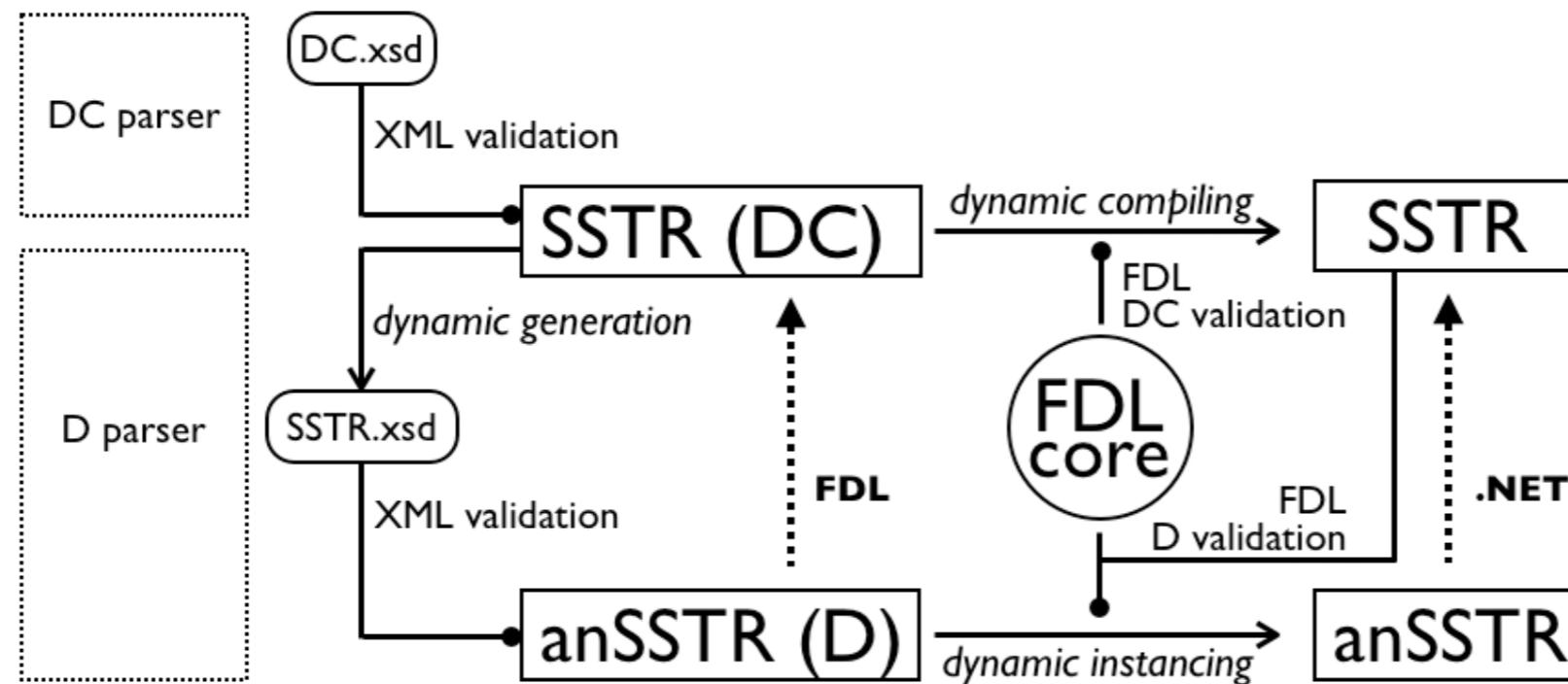
### Symbolic computing:

- Music followed-by Applauses => Performance
- Speech followed-by Applauses => Interview

## 4. FDL for PACE

### Example of an FDL descriptor:

➔ ShotSegmentationToolResult: validation of DC and Ds



- XML Schema syntactically constrains DCs and some aspects of Ds (a part of the logical structure, data types).
- FDL completes the validation process (the descriptor is a descendant from one of the three FDL generic ones, structures constraints are fulfilled...).

## 5. What is PACE gaining from FDL?

### What PACE is gaining from FDL:

➔ Semantically linked descriptors

The screenshot shows a Microsoft Internet Explorer browser window displaying a web page titled "Le grand échiquier". The page content includes a search bar with the text "4.1 La liste des émission qui contiennent ces segments" and a table of video sequences. The table has columns for ID, Durée, Résumé, and Type. Each row in the table includes a checkbox, a duration, a thumbnail image, and a "Voir" link. The "Type" column for all rows is "ITVW".

ID	Durée	Résumé	Type
<input type="checkbox"/> 002	00:19:41		ITVW <a href="#">Voir</a>
<input type="checkbox"/> 004	00:31:19		ITVW <a href="#">Voir</a>
<input type="checkbox"/> 006	00:08:11		ITVW <a href="#">Voir</a>
<input type="checkbox"/> 008	00:04:53		ITVW <a href="#">Voir</a>
<input type="checkbox"/> 010	00:16:48		ITVW <a href="#">Voir</a>
<input type="checkbox"/> 012	00:12:17		ITVW <a href="#">Voir</a>
<input type="checkbox"/> 014	00:10:15		ITVW <a href="#">Voir</a>

## **5. What is PACE gaining from FDL?**

### **What PACE is gaining from FDL:**

- ➔ Semantically linked descriptors
- ➔ Validation of extended descriptors
- ➔ Design consistent specific descriptors
- ➔ MPEG-7: no semantical link, no validation of extensions

### **Moreover:**

- ➔ FDL is fully modular: both tools and description classes are reusable.



## **6. Conclusion and future work**

### **Current state of PACE:**

- ➔ Every analysis tool works fine and produces FDL descriptions.
- ➔ The analysis graph has been manually validated.
- ➔ Automatic production of descriptions is not finished yet.

### **Future work:**

- ➔ Running PACE on other program collections.
- ➔ FDL: enhancing definition and control of structures (temporal and spatio-temporal constraints).