



## Deliverable D.6.1

### Month 12

# Software Specification of Video Databases with Querying Capabilities based on the Multimedia Content Description Standard MPEG7

<b>Project Start:</b>	01/02/2007
<b>Project Duration:</b>	36 Months
<b>Priority area</b>	2.3.6
<b>Contract No.:</b>	FP6-045547
<b>Website:</b>	

<b>Due-Date:</b>	16/01/2008
<b>Delivery:</b>	18/01/2008
<b>Lead Partner:</b>	CVC
<b>Project Leader</b>	UvA
<b>Dissemination Level:</b>	Public
<b>Status:</b>	Final
<b>Approved:</b>	
<b>Version:</b>	1.0



## Table of Contents

---

1. Introduction .....	3
2. Done Work .....	3
3. MPEG7.....	3
3.1. What is Mpeg7 .....	4
3.2. Mpeg7 java library.....	4
3.2.1. Implemented part.....	4
3.3. From NLG to Mpeg7 .....	8
3.4. Description of the implemented part of MPEG7 .....	11
3.4.1. The Header .....	11
3.4.2. The video path .....	11
3.4.3. Video Segment List.....	12
3.4.4. Video Segment .....	12
3.4.5. Video Segment description .....	12
3.4.6. Segment Time.....	12
3.4.7. MediaTimePoint.....	13
3.4.8. The MediaDuration .....	14
4. Software architecture .....	14
4.1. Software.....	15
4.1.1. Video Processing .....	15
4.1.2. Visual Analysis.....	15
4.1.3. Visual Classifier .....	15
4.1.4. Audio Analysis and Audio Classifier.....	15
4.1.5. Demo GUI.....	15
4.2. Data .....	15
4.2.1. VideoData .....	15
4.2.2. Shot/Audio segmentation.....	15
4.2.3. Visual/Audio features .....	16
4.2.4. Visual/Audio Concepts.....	16
4.2.5. Annotations .....	16
5. The Graphical User Interface (video data base system) .....	16
5.1. Import Video Data .....	16
5.2. Video Search Data .....	17
5.3. Predicates .....	20



## 1. Introduction

---

The objective of this work package is to consolidate and validate the textual, audio, speech, motion and visual features in a common representation and indexing structure to enable wide and simple usage by the other system components. The consolidation is split into two parts: one part contains the processing of video into shots, audio segments, and speech recognition; the other part of the consolidated software is the thesaurus of audiovisual detectors of semantic concepts.

Thus, on the one hand, the integrated system will be tested in the labs and evaluated with respect to the accuracy of both interpretation and retrieval, and the degree of interaction. On the other hand, since the representation of audiovisual features has to be used in the query ontology, the most effective way is to translate MPEG 7 descriptors in OWL so that they can easily be included and referred in the Query Ontology. In addition to translation of MPEG 7 visual descriptors in OWL this task will investigate methodologies to establish links and relationships between concepts defined in the query ontology and the visual descriptors so that query should be performed both on high semantic concepts and low level visual features

## 2. Done Work

---

The first year of the project we have focused on creating a prototype of the VidiVideo final software. We can divide the work in two main subjects: MPEG7, and a GUI (Graphical User Interface). In the following both are described in detail.

## 3. MPEG7

---

The first achievement that we had to accomplish is to learn MPEG-7, because we were not familiar with this language. This phase has been difficult because the information on MPEG-7 is still short and quite incomplete.



### **3.1. What is Mpeg7**

Mpeg7 is a multimedia content description standard. MPEG-7 is formally called Multimedia Content Description Interface. Thus, it is not a standard which deals with the actual encoding of moving pictures and audio, like MPEG-1, MPEG-2 and MPEG-4. It uses XML to store metadata.

It was designed to standardize:

- a set of Description Schemes (short DS in the standard) and Descriptors (short D in the standard)
- a language to specify these schemes, called the Description Definition Language (short DDL in the standard)
- a scheme for coding the description

### **3.2. Mpeg7 java library**

We have developed a java library to manage MPEG7 files. This library is written in Java and enables to load MPEG7 files to a java classes and vice-versa.

In short, we have developed java classes that store information of MPEG7. Furthermore, we have developed two conversion methods; one to load an xml MPEG7 file to instances of these java classes, and another one to store instances of these java classes to a file.

This library does only cover a very small part of MPEG7 –we have focused only on the parts of the standard that we are using- however, it has been designed so it can be extended in a very easy way.

#### **3.2.1. Implemented part**

As described above, not all MPEG7 specification has been implemented. Figure 1 illustrates which elements are currently done. In words, the implemented parts are:



1. The MediaLocator of the video, which indicates where is a video file.
2. Video TemporalDecomposition to indicate the shots of the video.
3. Inside the shots, we have implemented two types of text annotation
  - a. KeyWord annotation which stores annotations written in keywords
  - b. Sentence Annotation, which stores annotation written in a tree structured like a sentence.

Figure 2 is an example of an MPEG7 which can be used by the library. The example does not contain KeyWords.

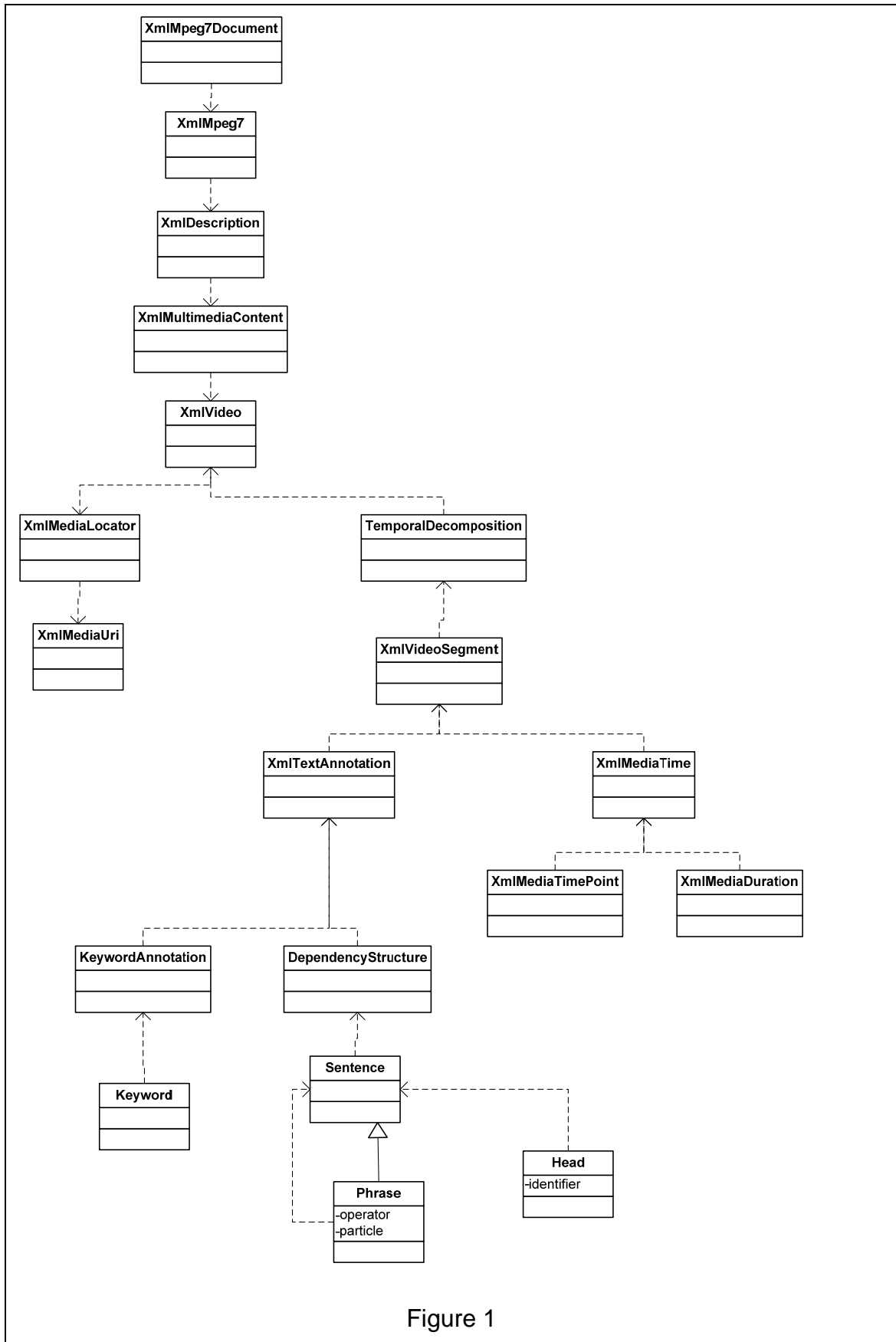


Figure 1



```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<Description xsi:type="ContentEntityType">
<MultimediaContent xsi:type="VideoType">
<Video id="TRECVID2006_1">
<MediaLocator>
<MediaUri>20051102_142800_LBC_NAHAR_ARB.mpg</MediaUri>
</MediaLocator>
<TemporalDecomposition gap="false" overlap="true">
<VideoSegment id="shot1_1_RKF">
<TextAnnotation relevance="0.44" confidence="1">
<DependencyStructure xml:lang="en">
<Sentence>

<!-- agent1 throws a red box to agent2: throws(agent1,
agent2, box, red) -->
<Phrase operator="subject">
<Head id="agent1"/>
</Phrase>
<Phrase operator="object">
<Phrase>
<Head id="red"/>
</Phrase>
<Head id="box"/>
</Phrase>
<Phrase functionWord="to">
<Head id="agent2"/>
</Phrase>
<Head id="throw"/>
</Sentence>
</DependencyStructure>
</TextAnnotation>
<MediaTime>
<MediaTimePoint>T00:00:00:0F30000</MediaTimePoint>
<MediaDuration>
PT00H00M00S1001N30000F</MediaDuration>
</MediaTime>
</VideoSegment>
</TemporalDecomposition>
</Video>
</MultimediaContent>
</Description>
</Mpeg7>
```

Figure 2

### 3.3. From NLG to Mpeg7

When the Mpeg7 library was developed, we needed to be able to convert from NLG - which is the format that is being used to store video data in CVC- to Mpeg7 files. Therefore, we have created a converter from these initial files to Mpeg7. Figure 3 shows an example of the current description file, and

Figure 4 indicates the corresponding Mpeg7 file using Keywords. The conversion to sentence like format has been started but is currently stopped (because it is not required).

```
470 ! pedestrian (Agent1)
470 ! appear (Agent1, upper_left)
492 ! walk (Agent1, upper_sidewalk)
583 ! turn (Agent1, right, upper_crosswalk)
591 ! stop (Agent1, upper_crosswalk)
615 ! object (Object1)
615 ! leave_object (Agent1, Object1)
630 ! pedestrian (Agent2)
630 ! appear (Agent2, upper_right)
642 ! walk (Agent2, upper_sidewalk)
656 ! walk (Agent1, upper_sidewalk)
687 ! abandoned_object (Object1, upper_crosswalk)
692 ! meet (Agent1, Agent2, upper_crosswalk)
800 ! vehicle (Agent3)
822 ! danger_of_runover (Agent3, Agent1)
825 ! stop (Agent1)
828 ! brake_up (Agent3)
828 ! danger_of_runover (Agent3, Agent2)
838 ! back_up (Agent2)
842 ! stop (Agent2)
852 ! accelerate (Agent3)
862 ! vehicle (Agent4)
862 ! appear (Agent4, left)
872 ! exit (Agent3, right)
873 ! chase (Agent1, Agent2)
```

Figure 3

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video>
        <MediaLocator>
          <MediaUri>E:/cvc/videoDeCVC/h264_vlc.mov</MediaUri>
        </MediaLocator>
```



```
<TemporalDecomposition>
  <VideoSegment>
    <TextAnnotation>
      <KeywordAnnotation>
        <Keyword>pedestrian</Keyword>
      </KeywordAnnotation>
    </TextAnnotation>
    <MediaTime>
<MediaTimePoint>T00:00:33:8F14</MediaTimePoint>
    </MediaTime>
  </VideoSegment>
  <VideoSegment>
    <TextAnnotation>
      <KeywordAnnotation>
        <Keyword>appear</Keyword>
      </KeywordAnnotation>
    </TextAnnotation>
    <MediaTime>
<MediaTimePoint>T00:00:33:8F14</MediaTimePoint>
    </MediaTime>
  </VideoSegment>
  <VideoSegment>
    <TextAnnotation>
      <KeywordAnnotation>
        <Keyword>walk</Keyword>
      </KeywordAnnotation>
    </TextAnnotation>
    <MediaTime>
<MediaTimePoint>T00:00:35:2F14</MediaTimePoint>
    </MediaTime>
  </VideoSegment>
  <VideoSegment>
    <TextAnnotation>
      <KeywordAnnotation>
        <Keyword>turn</Keyword>
      </KeywordAnnotation>
    </TextAnnotation>
    <MediaTime>
<MediaTimePoint>T00:00:41:9F14</MediaTimePoint>
    </MediaTime>
  </VideoSegment>
  <VideoSegment>
    <TextAnnotation>
      <KeywordAnnotation>
        <Keyword>stop</Keyword>
      </KeywordAnnotation>
    </TextAnnotation>
    <MediaTime>
<MediaTimePoint>T00:00:42:3F14</MediaTimePoint>
    </MediaTime>
  </VideoSegment>
  <VideoSegment>
    <TextAnnotation>
      <KeywordAnnotation>
```



```
        <Keyword>object</Keyword>
      </KeywordAnnotation>
    </TextAnnotation>
    <MediaTime>
<MediaTimePoint>T00:00:43:13F14</MediaTimePoint>
    </MediaTime>
  </VideoSegment>
<VideoSegment>
  <TextAnnotation>
    <KeywordAnnotation>
      <Keyword>leave_object</Keyword>
    </KeywordAnnotation>
  </TextAnnotation>
  <MediaTime>
<MediaTimePoint>T00:00:43:13F14</MediaTimePoint>
    </MediaTime>
  </VideoSegment>
<VideoSegment>
  <TextAnnotation>
    <KeywordAnnotation>
      <Keyword>pedestrian</Keyword>
    </KeywordAnnotation>
  </TextAnnotation>
  <MediaTime>
<MediaTimePoint>T00:00:45:0F14</MediaTimePoint>
    </MediaTime>
  </VideoSegment>
<VideoSegment>
  <TextAnnotation>
    <KeywordAnnotation>
      <Keyword>appear</Keyword>
    </KeywordAnnotation>
  </TextAnnotation>
  <MediaTime>
<MediaTimePoint>T00:00:45:0F14</MediaTimePoint>
    </MediaTime>
  </VideoSegment>
<VideoSegment>
  <TextAnnotation>
    <KeywordAnnotation>
      <Keyword>walk</Keyword>
    </KeywordAnnotation>
  </TextAnnotation>
  <MediaTime>
<MediaTimePoint>T00:00:45:12F14</MediaTimePoint>
    </MediaTime>
  </VideoSegment>
<VideoSegment>
  <TextAnnotation>
    <KeywordAnnotation>
      <Keyword>walk</Keyword>
    </KeywordAnnotation>
  </TextAnnotation>
  <MediaTime>
```

```
<MediaTimePoint>T00:00:46:12F14</MediaTimePoint>
  </MediaTime>
</VideoSegment>
<VideoSegment>
  <TextAnnotation>
    <KeywordAnnotation>
      <Keyword>chase</Keyword>
    </KeywordAnnotation>
  </TextAnnotation>
  <MediaTime>
<MediaTimePoint>T00:01:02:5F14</MediaTimePoint>
  </MediaTime>
</VideoSegment>
</TemporalDecomposition>
</Video>
</MultimediaContent>
</Description>
</Mpeg7>
```

Figure 4

### 3.4. Description of the implemented part of MPEG7

In the following we describe, briefly, the MPEG7 implemented part.

#### 3.4.1. The Header

All files should start with:

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video>
```

Inside the video tag, all the content is described.

#### 3.4.2. The video path

In order to store where the video, that the MPEG7 is describing, can be found, we should insert the following tag as the first child of the video tag –where sample.avi must be changed to the corresponding path-

```
<MediaLocator> <MediaUri>sample.avi</MediaUri>
</MediaLocator>
```

### 3.4.3. Video Segment List

After the MediaLocator, <TemporalDecomposition> should be the next child of video. Inside this tag, the shots are described.

```
<TemporalDecomposition>
```

### 3.4.4. Video Segment

Inside the TemporalDecomposition a Video Segments are listed. The video segments contain the time when the segment starts, and the description of the content.

```
<VideoSegment>
```

### 3.4.5. Video Segment description

The first child of the VideoSegment is the description of the data of the segment. In the following we expose an example of the text annotation. In case that there is more than one keyword, they should be listed after the first one.

```
<TextAnnotation>  
  <KeywordAnnotation>  
    <Keyword>plane</Keyword>  
  </KeywordAnnotation>  
</TextAnnotation>
```

### 3.4.6. Segment Time

The second child of VideoSegment indicates when the video starts, and which is the duration of the segment.

```
<MediaTime>  
<MediaTimePoint>T00:00:46:12F14</MediaTimePoint>  
<MediaDuration>PT00H00M00S1001N30000F</MediaDuration>  
</MediaTime>
```

In the following we describe how MediaTimePoint is written



### 3.4.7. MediaTimePoint

MediaTimePoint describes a time stamp of the media using Gregorian date and day time without specifying the TZD.

The format is YYYY-MM-DDThh:mm:ss:nnnFNNN. Where following lexicals are used for digits of the corresponding date/time elements:

- Y: Year, can be a variable number of digits,
- M: Month,
- D:Day,
- h: hour,
- m: minute,
- s: second,
- n: number of fractions, nnn can be any number between 0 and NNN-1 (NNN and with it nnn can have an arbitrary number of digits).
- N: number of fractions of one second which are counted by nnn. NNN can have a arbitrary number of digits and is not limited to three.

Also delimiters for the time specification (T) and the number of fractions of one second are used (F).

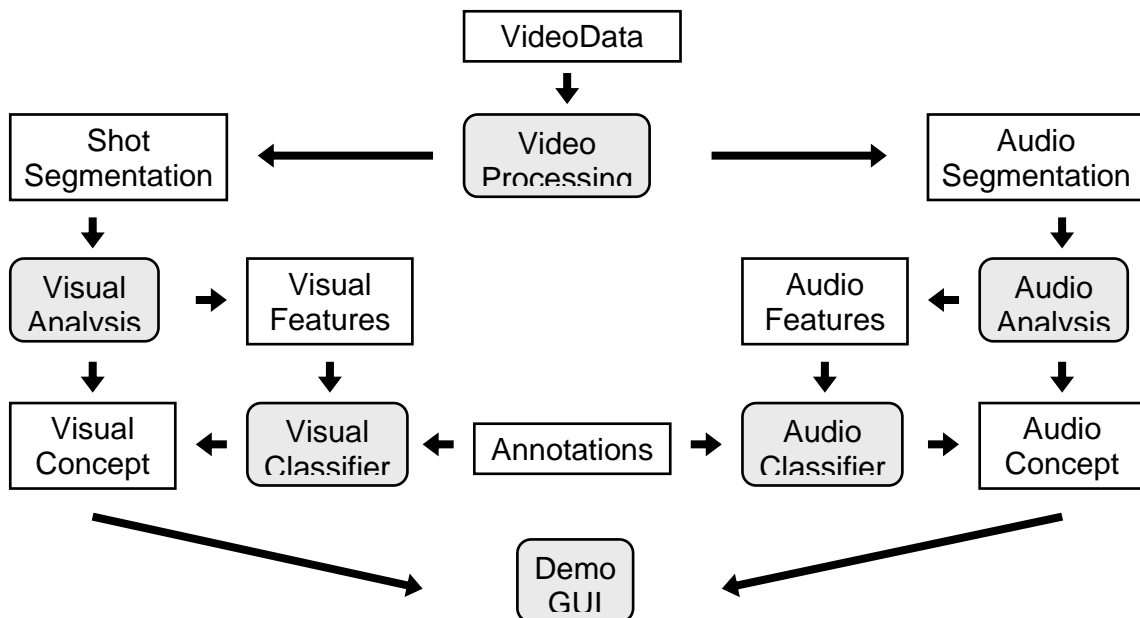
Beside the specification of Gregorian Date and day time, counting the number (nnn) of fractions (FNNN) of a second can allow a higher precision than one second. If counting fractions is used, the number of fractions making up one second (FNNN) shall be specified along with the counted number of fraction (nnn). Thus the FNNN defines the value range of the counted number of fractions (nnn) value. So the value range of 'nnn' is limited from 0 to FNNN-1.

### 3.4.8. The MediaDuration

Describes the duration of a time interval according to days and day time of a notion of time encoded in the media without specifying a difference in the TZD. The time interval is defined as a half open time interval with the closed end being at the beginning. A simpleType representing a duration in time using a lexical representation of days (nD), time duration and a fraction specification (TnHnMnSnN) including the specification of the number of fractions of one second (nF): PnDTnHnMnSnNnF.

## 4. Software architecture

In order to integrate the software, a software architecture has been designed. The following diagram defines the software architecture of the application. The grey rounded shaped rectangles represent software units, and rectangular white shaped rectangles represent data transferred between these software units. The data, in exception of VideoData, will be stored in an MPEG7 form.





## **4.1. Software**

### **4.1.1. Video Processing**

The first one is the Video Processing. This software receives, as input, videos, and outputs the shot segmentation and the audio segmentation for this video.

### **4.1.2. Visual Analysis**

This software, using the shot segmentation and the video, obtains visual features from the shots. Moreover, applying this features to the classifiers created by the Visual Classifier, generates visual concepts for the video shots.

### **4.1.3. Visual Classifier**

This software, given annotations (ground truth) from one video, generates classifiers for new videos. To do so, it uses the visual features (for the annotated videos) created by the visual analysis.

### **4.1.4. Audio Analysis and Audio Classifier**

This software is the equivalent part of Visual Analysis and Visual Classifier, for audio features.

### **4.1.5. Demo GUI**

Finally a Graphic User Interface is developed to be able to search into the videos with the visual and audio concepts that have been generated by the previous software.

## **4.2. Data**

### **4.2.1. VideoData**

This are the videos to be used

### **4.2.2. Shot/Audio segmentation**

MPEG7 file with the shots of a video



#### **4.2.3. Visual/Audio features**

MPEG7 file with the low level features of the video shots

#### **4.2.4. Visual/Audio Concepts**

MPEG7 file with visual concepts that appear in each shot of the video

#### **4.2.5. Annotations**

MPEG7 files with the same format as Visual/Audio Concepts, but with real information (ground truth).

## **5. The Graphical User Interface (video data base system)**

---

Finally we have developed a GUI where MPEG7 files are introduced, in order to be queried, by searching keywords, in a graphical user-friendly style. Therefore, this software is not only a GUI, but a video data base, with searching capabilities. The software is still being developed, but a first working version has been released.

The software consists in two parts:

1. Import video data
2. Video data search

### **5.1. Import Video Data**

The software maintains information of videos in a data base. Therefore, an importing system has been developed to import data to the DB. Using the GUI, the user can load an MPEG7 to this DB. Any MPEG7 file similar to the one on

Figure 4 can be loaded to the system.



It is important to notice that the data is stored in a database, not in files. By doing this –among others-, you may have many applications (users) acceding to the same database.

## **5.2. Video Search Data**

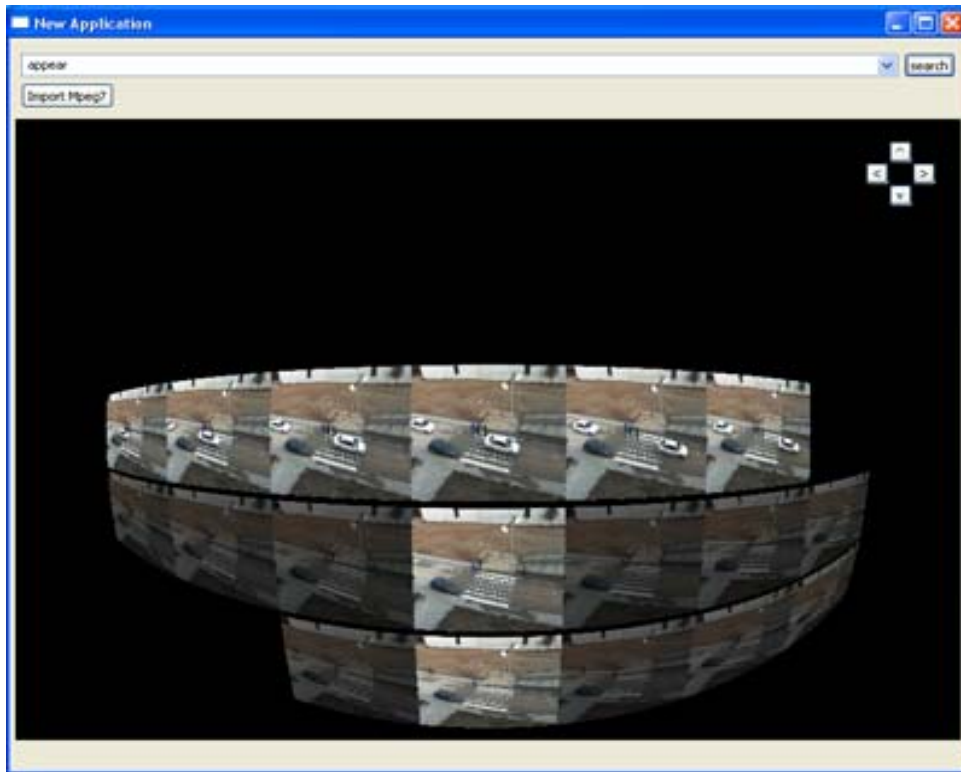
When data has been imported to the DB, it can be consulted via the GUI. To do so, the user must introduce which keyword he is looking for, and the system will show the shots where this keyword appears. For user facility, not only a picture of the shot is displayed, but also some neighbour shots. More precisely, when a search is done, the user will be presented with a kind of sphere with some images putted in a matrix form. Each row represents a video segment where the desired concept is found. In each row, some images of the video are shown, in order to see what is happening in time.

From the Mpeg7 we are using only a small part. In general terms we are using:

1. The MediaLocator of the video, which indicates where is a video file.
2. VideoSegmentation to indicate the shots of the video.
  - a. Inside the shots, KeyWord annotation is used.

It is important to notice that the segments, in this case, are not complete (not all the video is inside a segment), and that they are overlapped. Moreover, we use

all types of segments may be



used.

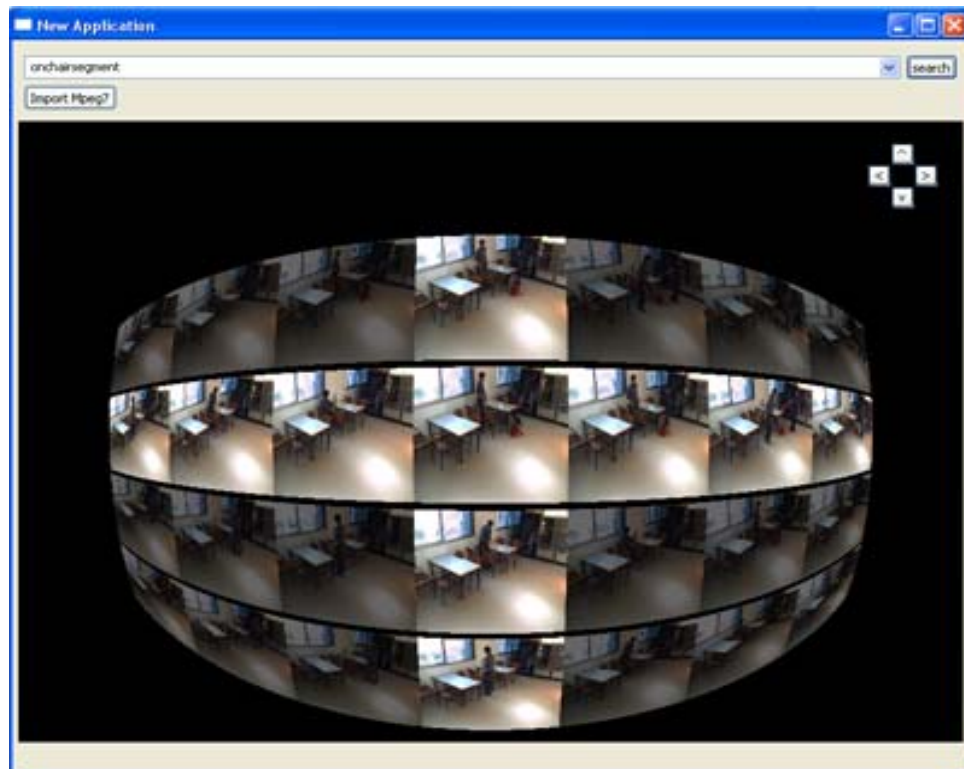
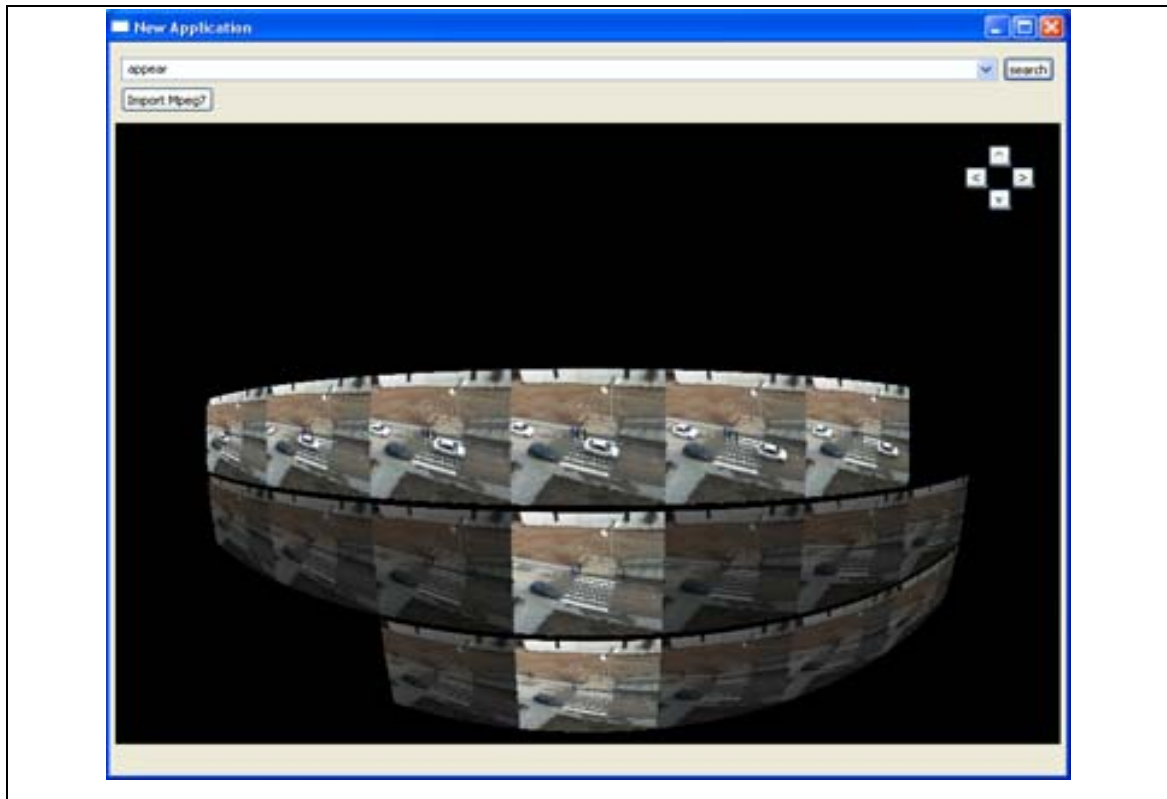


Figure 5 contains two screenshots of a search in the search engine, the first one of the keyword appear, and the second one of the keyword “on chair segment”. The rows are different shots where the searched keyword appears, and the columns are the neighbour shots.



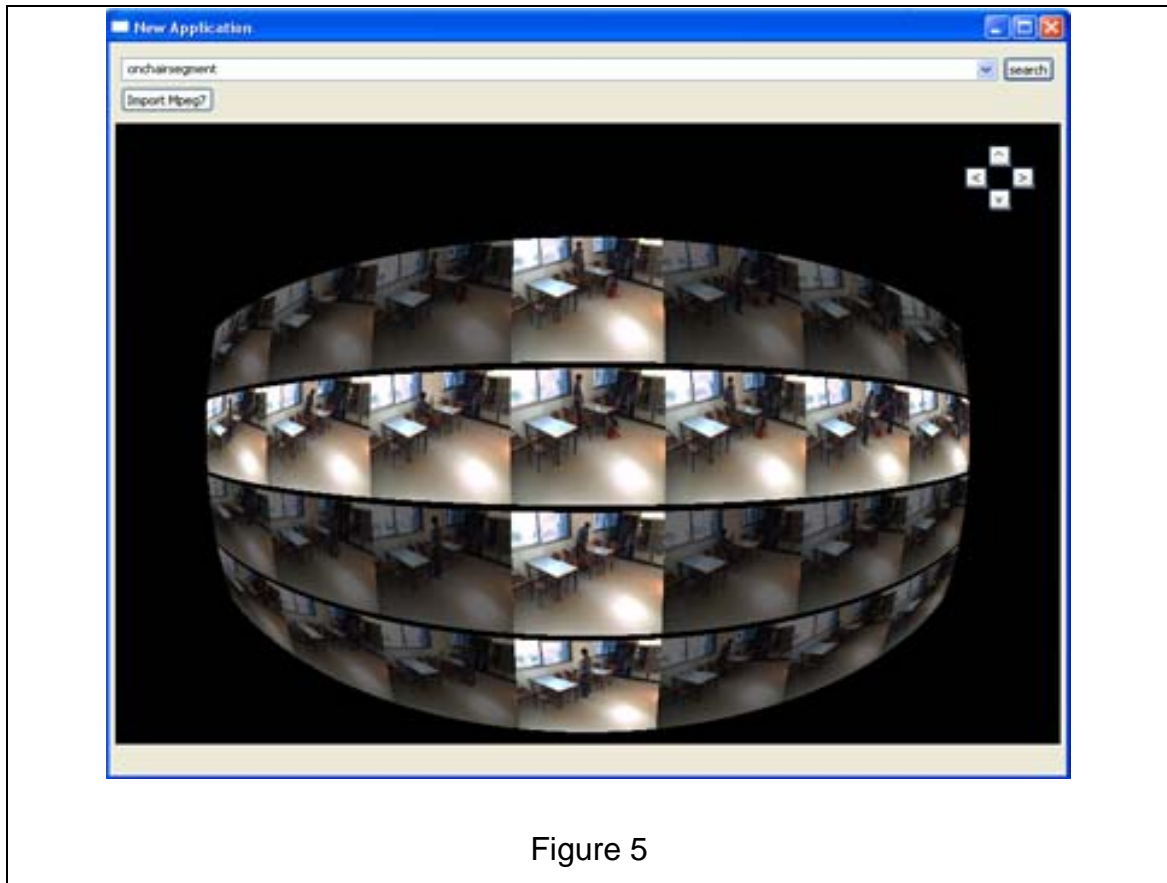


Figure 5

### 5.3. *Predicates*

When a user wants to make a search, he should chose from a list of predicates. This list depends on the MPEG7 files that have been imported; therefore the user will be able to chose a keyword if (and only if) an MPEG7 that contains this keyword has been introduced to the system –the file has been imported-.

Currently the list of predicates that can be searched for are:

- pedestrian
- appear
- walk
- turn
- stop
- object
- leave\_object
- pedestrian
- appear
- walk
- walk
- abandoned\_object



- meet
- vehicle
- danger\_of\_runover
- stop
- brake\_up
- danger\_of\_runover
- back\_up
- stop
- accelerate
- vehicle
- appear
- exit
- chase
- on cafeteria segment
- on vending segment
- stopped at the vending machine
- on chair segment
- sitting on chair
- on\_sideway\_seg
- agent\_walking\_on\_the\_waiting\_line
- on\_crosswalk
- crossed
- on\_the\_other\_sidewalk
- stopped\_in\_the\_waiting\_line
- on\_road