

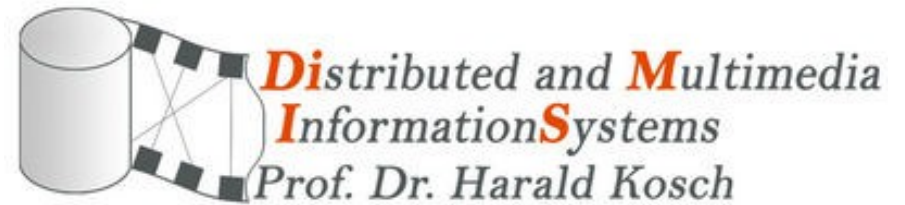
# Soutenance de stage

Projet de fin d'études



# INTRODUCTION

- ▶ stage d'une durée de cinq mois
- ▶ réalisé au DIMIS, à l'Université de Passau (Allemagne)
- ▶ proposé par le Dr Vanessa El-Khoury



*Le stage s'inscrit dans la continuité du travail du Dr El-Khoury qui a développé PIAF, un framework d'adaptation personnalisée de vidéos.*

# CONTEXTE

## *Personalized video Adaptation Framework*

### Exemples de contraintes :

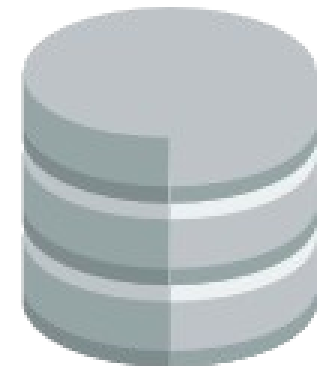
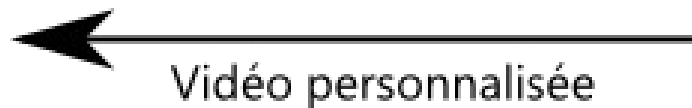
- ▶ pas de placement de produits
- ▶ pas de violence
- ▶ pas d'alcool

### Exemples d'adaptation :

- ▶ flouter les images
- ▶ supprimer les images
- ▶ couper le plan tout entier



Utilisateur



PIAF

# CONTEXTE

## Annotation de vidéos



### Informations de structure :

- ▶ découpage de la vidéo en scènes
- ▶ découpage des scènes en plans



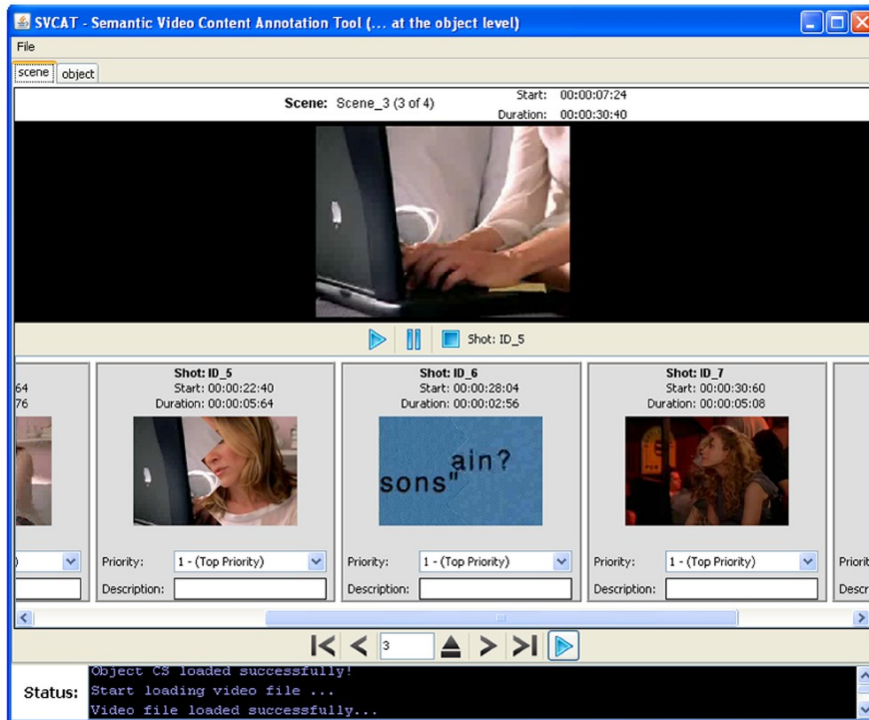
### Informations sémantiques :

- ▶ position spatio-temporelle d'un objet
- ▶ valeur sémantique associée à l'objet
- ▶ valeur sémantique associée à une scène / à un plan

# OBJECTIFS

## S.V.C.A.T.

Logiciel d'annotation à l'état de prototype



### Rendre le prototype fonctionnel et indépendant :

- ▶ implémenter la lecture de flux vidéo
- ▶ détecter automatiquement les plans
- ▶ exporter le résultat en XML

### Améliorer l'outil :

- ▶ corriger le suivi temporel des objets
- ▶ gérer plus de cas particuliers pour les objets
- ▶ réorganiser le code afin d'être indépendant des formats utilisés
- ▶ revoir et simplifier l'interface

# OBJECTIFS

## Spécifications sur le contenu :

- ▶ standard MPEG-7 pour les schémas de classification
- ▶ standard MPEG-7 pour le fichier de sortie
- ▶ contours des objets précis au pixel près

## Spécifications sur la forme :

- ▶ programme en Java
- ▶ programme portable

```
<mpeg7:VideoSegment>
  <mpeg7:TemporalMask>
    <mpeg7:SubInterval>
      <mpeg7:MediaTimePoint>T00:00:00:000F1000</mpeg7:Media
      <mpeg7:MediaDuration>PT0.33366667S</mpeg7:MediaDurati
    </mpeg7:SubInterval>
  </mpeg7:TemporalMask>
  <mpeg7:SpatioTemporalDecomposition/>
</mpeg7:VideoSegment>
<mpeg7:VideoSegment>
  <mpeg7:TemporalMask>
    <mpeg7:SubInterval>
      <mpeg7:MediaTimePoint>T00:00:00:000F1000</mpeg7:Media
      <mpeg7:MediaDuration>PT0.33366667S</mpeg7:MediaDurati
    </mpeg7:SubInterval>
  </mpeg7:TemporalMask>
  <mpeg7:SpatioTemporalDecomposition/>
</mpeg7:VideoSegment>
<mpeg7:VideoSegment>
  <mpeg7:TemporalMask>
    <mpeg7:SubInterval>
      <mpeg7:MediaTimePoint>T00:00:00:000F1000</mpeg7:Media
      <mpeg7:MediaDuration>PT0.33366667S</mpeg7:MediaDurati
    </mpeg7:SubInterval>
  </mpeg7:TemporalMask>
  <mpeg7:SpatioTemporalDecomposition/>
</mpeg7:VideoSegment>
```

# IMPLÉMENTATION

## **Modélisation des parties du programme :**

- ▶ image / vidéo
- ▶ schémas de classification
- ▶ annotations et contours

## **Développement des outils et des algorithmes :**

- ▶ décodage vidéo
- ▶ importation / exportation des schémas de classification
- ▶ détection automatique des plans
- ▶ détection de contours
- ▶ suivi de contours

## **Création d'une interface graphique**

# IMPLÉMENTATION

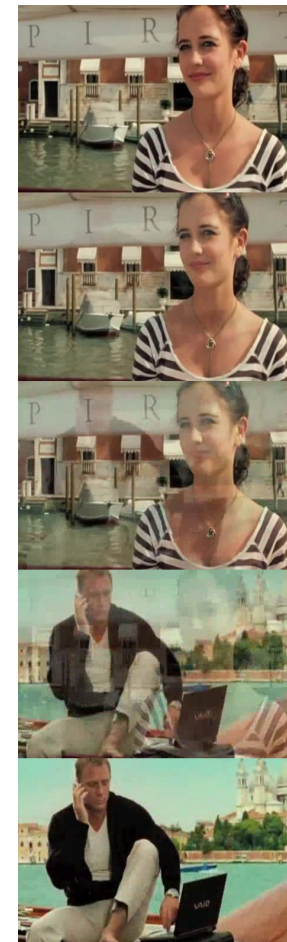
## Détection des plans

**Les transitions entre les plans sont très variées :**

- ▶ coupure sèche
- ▶ fondu au noir
- ▶ fondu enchaîné
- ▶ balayage
- ▶ rabat et couverture
- ▶ ...

**Trouver manuellement les transitions est long :**

- ▶ il faut parcourir les *frames* une par une
- ▶ il peut y avoir beaucoup de plans à traiter
- ▶ il est difficile de trouver la limite dans certains cas





# IMPLÉMENTATION

## Détection des plans

La détection automatique des plans simplifie le travail de l'annotateur.

### Plusieurs difficultés de mise en œuvre :

- ▶ chaque type de transition a ses propres caractéristiques
- ▶ la détection dépend de plusieurs autres algorithmes
- ▶ il est indispensable de pouvoir accéder à n'importe quelle *frame* de la vidéo

Transition	Valeur(s) caractéristique(s)
Coupure sèche	Différence d'histogrammes <i>Edge change ratio</i>
Fondu au noir	Écart type de l'intensité des pixels
Fondu enchaîné	<i>Edge-based contrast</i>

# IMPLÉMENTATION

## Détection des plans

### Algorithmes requis :

- ▶ détection de bords
- ▶ mise en cache des résultats intermédiaires
- ▶ calcul des seuils des valeurs caractéristiques



### Limites de la détection des plans :

- ▶ l'algorithme est long à exécuter et monopolise les ressources de l'ordinateur
- ▶ l'implémentation n'est pas complète
- ▶ il y a parfois des erreurs dans la détection

Malgré ses défauts, l'outil reste quand même **un élément indispensable du programme.**

# IMPLÉMENTATION

## Détection d'un contour

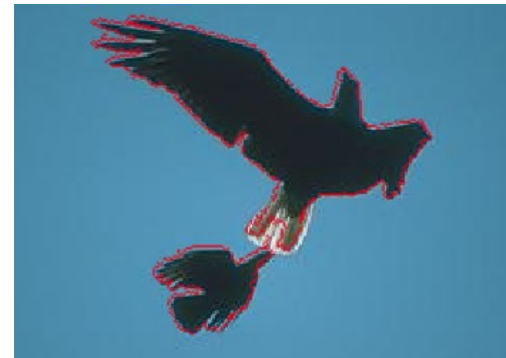
Dessiner un contour exact à la main est généralement soit **long**, soit **peu précis**.

**On utilise un algorithme de contour actif :**

- ▶ l'annotateur entoure grossièrement l'objet
- ▶ l'algorithme adapte le contour un certain nombre de fois
- ▶ l'annotateur peut continuer d'itérer jusqu'à être satisfait

**Améliorations envisagées :**

- ▶ apprentissage à partir des annotations précédentes
- ▶ détection complètement automatique à partir d'une base de données d'objets



# IMPLÉMENTATION

## Suivi temporel d'un contour

Le suivi temporel d'un contour permet d'éviter d'avoir à détecter le contour d'un objet sur toutes les *frames* où il apparaît.

Il s'agit toujours de **faciliter le travail d'annotation**.

**Plusieurs implémentations sont possibles :**

- ▶ modèle à états
- ▶ minimisation d'une fonction d'énergie
- ▶ approche naïve



# IMPLÉMENTATION

## Suivi temporel d'un contour

### **Principe de l'algorithme naïf :**

- ▶ l'utilisateur applique la détection automatique du contour sur un objet
- ▶ le contour entoure grossièrement l'objet à la *frame* suivante
- ▶ on applique à nouveau l'algorithme du contour actif et ainsi de suite

### **Avantages :**

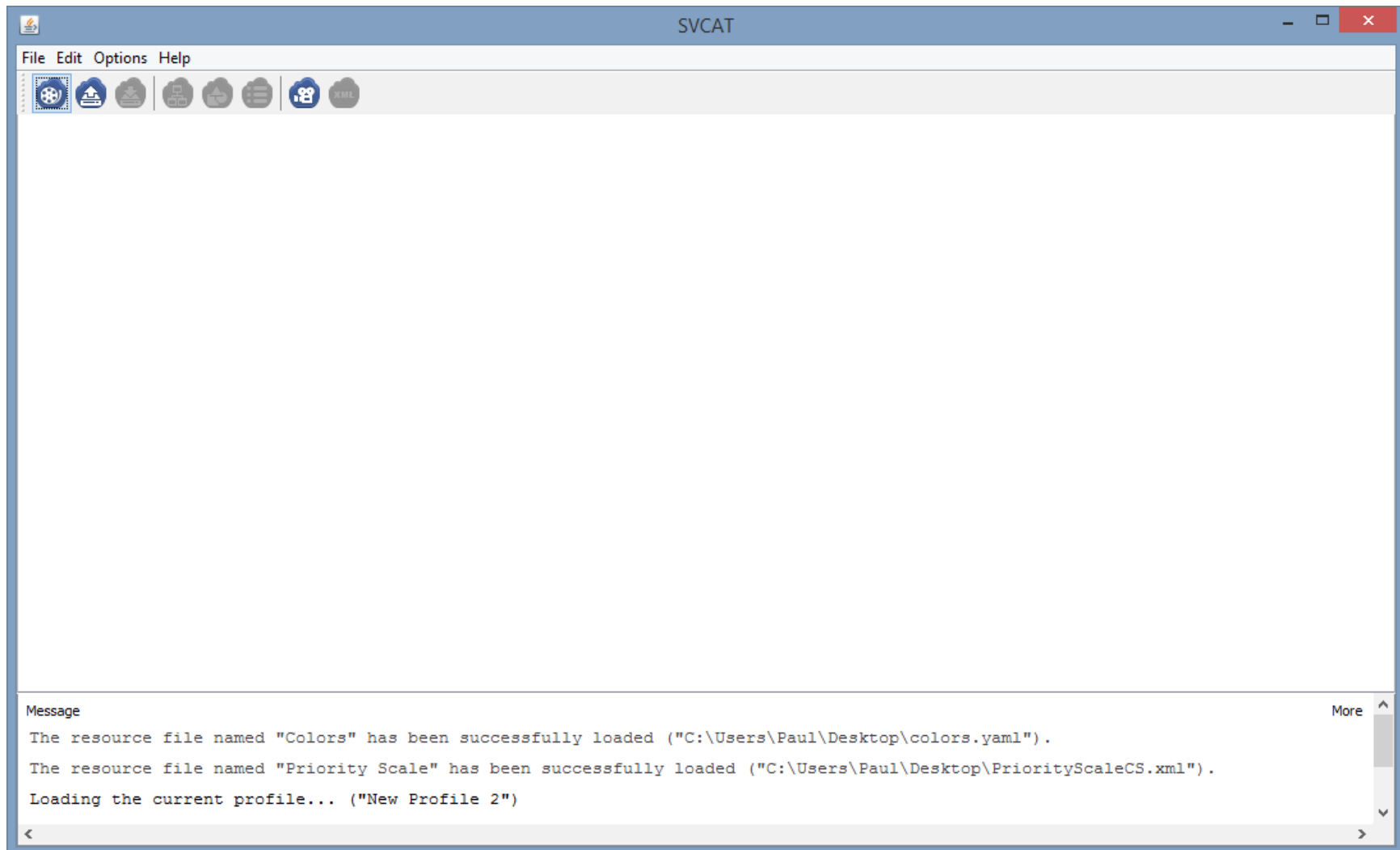
- ▶ suffisamment précis
- ▶ relativement rapide
- ▶ simple à mettre en place

# EXEMPLE

Structure de la vidéo

Annotation des objets et des plans

# EXAMPLE



# EXEMPLE

## Structure de la vidéo

The screenshot displays the SVCAT software interface. At the top, there is a menu bar with 'File', 'Edit', 'Options', and 'Help'. Below the menu is a toolbar with icons for scene creation, separator insertion, separator removal, merge, and split. The main workspace is divided into two panels: 'Structure' on the left, which shows a filmstrip with three video frames, and 'Shot' on the right, which contains a form for defining shot parameters. At the bottom, a 'Message' log shows the status of the document save operation.

**Structure**

**Shot**

First frame's index:	<input type="text" value="0"/>
First frame's timestamp:	<input type="text" value="00:00:00.000"/>
Last frame's index:	<input type="text" value="2088"/>
Last frame's timestamp:	<input type="text" value="00:01:09.669"/>
Number of frames:	<input type="text" value="2089"/>
Duration:	<input type="text" value="00:01:09.702"/>
First shot of the scene:	<input type="text" value="yes"/>
Last shot of the scene:	<input type="text" value="yes"/>
Transition before:	<input type="text" value="UNDEFINED"/>
Transition after:	<input type="text" value="UNDEFINED"/>

**Message**

```
The SVCAT document has been successfully saved ("C:\Users\Paul\Desktop\Exemple.svcat").  
Saving the SVCAT document "C:\Users\Paul\Desktop\Exemple.svcat"..  
The SVCAT document has been successfully saved ("C:\Users\Paul\Desktop\Exemple.svcat").
```



# EXEMPLE

## Structure de la vidéo

The screenshot displays the SVCAT software interface. At the top, there is a menu bar with 'File', 'Edit', 'Options', and 'Help'. Below the menu is a toolbar with icons for various functions. The main workspace is divided into two sections: 'Structure' and 'Shot'.

**Structure:** This section shows a grid of video frames arranged in a filmstrip format. The frames are organized into rows and columns, representing different shots and scenes. The top row shows a man sitting at a table, and the second row shows a woman talking on a phone. The third row shows a man looking at a phone, and the fourth row shows a man looking at a phone. The fifth row shows a man looking at a phone, and the sixth row shows a man looking at a phone.

**Shot:** This section provides detailed information for the selected shot. The information is as follows:

First frame's index:	289
First frame's timestamp:	00:00:09.642
Last frame's index:	549
Last frame's timestamp:	00:00:18.318
Number of frames:	261
Duration:	00:00:08.708
First shot of the scene:	no
Last shot of the scene:	no
Transition before:	HARD
Transition after:	HARD

**Message:** A message box at the bottom of the interface displays the following text:

```
The SVCAT document has been successfully saved ("C:\Users\Paul\Desktop\Exemple.svcat").  
Saving the SVCAT document "C:\Users\Paul\Desktop\Exemple.svcat"..  
The SVCAT document has been successfully saved ("C:\Users\Paul\Desktop\Exemple.svcat").
```

# EXEMPLE

## Structure de la vidéo

The screenshot displays the SVCAT software interface. At the top, there is a menu bar with 'File', 'Edit', 'Options', and 'Help'. Below the menu is a toolbar with various icons. The main workspace is divided into two main sections: 'Structure' and 'Shot'.

**Structure:** This section shows a grid of video frames arranged in a sequence. The frames are organized into rows and columns, representing different shots and scenes. The frames show a man in a suit sitting at a table, holding a smartphone, and looking at the screen. The frames are arranged in a sequence that suggests a narrative flow.

**Shot:** This section provides detailed information about the selected shot. The information is organized into a table-like structure with labels and input fields:

Label	Value
First frame's index:	1062
First frame's timestamp:	00:00:35.435
Last frame's index:	1134
Last frame's timestamp:	00:00:37.837
Number of frames:	73
Duration:	00:00:02.435
First shot of the scene:	no
Last shot of the scene:	yes
Transition before:	HARD
Transition after:	FADE

At the bottom of the interface, there is a 'Message' section with a scrollable area. It contains the following text:

```
The SVCAT document has been successfully saved ("C:\Users\Paul\Desktop\Exemple.svcat").  
Saving the SVCAT document "C:\Users\Paul\Desktop\Exemple.svcat"..  
The SVCAT document has been successfully saved ("C:\Users\Paul\Desktop\Exemple.svcat").
```

# EXEMPLE

## Annotation des objets et des plans

The screenshot displays the SVCAT software interface, which is used for video annotation. The window title is "SVCAT". The menu bar includes "File", "Edit", "Options", and "Help". The toolbar contains icons for various functions, including a gear, a folder, a document, a camera, a magnifying glass, a list, a trash, and an XML icon.

The main interface is divided into several panels:

- Video structure:** A tree view showing a hierarchy of scenes and shots. "Scene 2" is expanded, showing "Shot 2" through "Shot 10". "Shot 4" is currently selected and highlighted in blue.
- Segment annotation:** A large empty white box for adding annotations to video segments. Below it are buttons for "Edit", "Delete", "Add free-text", and "Add URI".
- Contour editor:** A central video player showing a scene with a man on a boat. Above the video is a toolbar with shapes: "Rectangle" (selected), "Ellipse", "Polygon", "Snake", "Track", and "Delete". Below the video is a timeline with a playhead at 00:00:18.351 and a frame number of 550. Playback controls (back, pause, play, forward) and buttons for "Set start point", "Split", and "Set end point" are visible.
- Object annotation:** A large empty white box for adding object annotations. Below it are buttons for "Edit", "Delete", "Merge", and "Separate".

At the bottom of the window is a "Message" log with the following text:

```
The SVCAT document has been successfully saved ("C:\Users\Paul\Desktop\Exemple.svcat").  
Saving the SVCAT document "C:\Users\Paul\Desktop\Exemple.svcat"....  
The SVCAT document has been successfully saved ("C:\Users\Paul\Desktop\Exemple.svcat").  
Saving the SVCAT document "C:\Users\Paul\Desktop\Exemple.svcat"....
```

# EXEMPLE

## Annotation des objets et des plans



# EXEMPLE

## Annotation des objets et des plans

The screenshot displays the SVCAT software interface, which is used for video annotation. The main window is titled "SVCAT" and features a menu bar with "File", "Edit", "Options", and "Help". Below the menu bar is a toolbar with various icons for file operations and editing.

The interface is divided into several panels:

- Video structure:** A tree view on the left showing a hierarchy of scenes and shots. "Scene 2" is expanded, showing "Shot 2" through "Shot 10". "Shot 4" is currently selected and highlighted in blue.
- Segment annotation:** A large empty white box for adding text annotations to video segments. Below it are buttons for "Edit", "Delete", "Add free-text", and "Add URI".
- Contour editor:** A central video player showing a scene with a man on a boat. Above the video is a toolbar with shapes for "Rectangle", "Ellipse", and "Polygon", and tools for "Snake", "Track", and "Delete". Below the video is a timeline with a playhead at 550 and a time display of 00:00:18.351. Playback controls (back, pause, play, forward) and buttons for "Set start point", "Split", and "Set end point" are located at the bottom of this panel.
- Object annotation:** A panel on the right showing a list of annotated objects. One object is highlighted in blue: "Snake [ 550 ; 708 ]" with a yellow square icon. Below the list are buttons for "Edit", "Delete", "Merge", and "Separate".

At the bottom of the window is a "Message" panel with a scrollable text area. It contains the following text:

```
The SVCAT document has been successfully saved ("C:\Users\Paul\Desktop\Exemple.svcat").  
Saving the SVCAT document "C:\Users\Paul\Desktop\Exemple.svcat"..  
The SVCAT document has been successfully saved ("C:\Users\Paul\Desktop\Exemple.svcat").
```

# EXEMPLE

## Annotation des objets et des plans

The screenshot displays the SVCAT software interface, which is used for video annotation. The window title is "SVCAT". The menu bar includes "File", "Edit", "Options", and "Help". The toolbar contains icons for various functions, including a camera, a folder, a document, a trash can, a list, a paw print, and an XML icon.

The main interface is divided into several panels:

- Video structure:** A tree view showing a hierarchy of scenes and shots. "Scene 2" is expanded, showing "Shot 2" through "Shot 10". "Shot 4" is currently selected and highlighted in blue.
- Segment annotation:** A large empty white box for adding text annotations to video segments. Below it are buttons for "Edit", "Delete", "Add free-text", and "Add URI".
- Contour editor:** A central video player showing a scene of a man on a boat. Above the video are tools for drawing shapes: "Rectangle", "Ellipse", "Polygon", "Snake", "Track", and "Delete". Below the video is a timeline with a playhead at 00:00:22.689. Playback controls (back, pause, play, forward) and buttons for "Set start point", "Split", and "Set end point" are visible.
- Object annotation:** A panel on the right showing a list of annotated objects. One object is highlighted in blue: "Dynamic [ 550 ; 702 ]" with a URI "urn:ProductPlacementCS:Computer:VAI...". Below the list are buttons for "Edit", "Delete", "Merge", and "Separate".

At the bottom of the window is a "Message" log with the following text:

```
The SVCAT document has been successfully saved ("C:\Users\Paul\Desktop\Exemple.svcat").  
Saving the SVCAT document "C:\Users\Paul\Desktop\Exemple.svcat"..  
The SVCAT document has been successfully saved ("C:\Users\Paul\Desktop\Exemple.svcat").
```

# EXEMPLE

## Annotation des objets et des plans

The screenshot displays the SVCAT software interface, which is used for video annotation. The main window is titled "SVCAT" and features a menu bar with "File", "Edit", "Options", and "Help". Below the menu bar is a toolbar with various icons for file operations and editing.

The interface is divided into several panels:

- Video structure:** A tree view showing a hierarchy of scenes and shots. "Scene 2" is expanded, showing shots from "Shot 2" to "Shot 10". "Shot 4" is currently selected and highlighted in blue.
- Segment annotation:** A panel for editing the selected segment. It shows a "URI [ 550 ; 708 ]" and a "URI: urn:PriorityScaleCS:2". Below this are buttons for "Edit", "Delete", "Add free-text", and "Add URI".
- Contour editor:** A panel for defining shapes and tracks. It includes tools for "Rectangle", "Ellipse", "Polygon", "Snake", "Track", and "Delete". A video player is embedded in this panel, showing a man sitting on a boat. A red line indicates the current position in the video, with a timecode of "00:00:22.689" and a frame number of "680". Below the video player are playback controls (back, pause, play, forward) and buttons for "Set start point", "Split", and "Set end point".
- Object annotation:** A panel for defining object annotations. It shows a "Dynamic [ 550 ; 702 ]" annotation with a URI: "urn:ProductPlacementCS:Computer:VAI". Below this are buttons for "Edit", "Delete", "Merge", and "Separate".

At the bottom of the interface is a "Message" panel, which displays the following text:

```
The SVCAT document has been successfully saved ("C:\Users\Paul\Desktop\Exemple.svcat").  
Saving the SVCAT document "C:\Users\Paul\Desktop\Exemple.svcat"..  
The SVCAT document has been successfully saved ("C:\Users\Paul\Desktop\Exemple.svcat").
```

# PRINCIPAUX OUTILS

## **Apache MAVEN :**

- ▶ exécute toutes les étapes de la compilation d'un projet Java
- ▶ installe toutes les dépendances
- ▶ génère la Javadoc

## **JAXB (*Java Architecture for XML Binding*) :**

- ▶ crée les classes Java à partir de schémas XSD
- ▶ facilite la lecture et l'écriture de fichiers XML

## **Google Guava :**

- ▶ ajoute de nouvelles collections
- ▶ propose un système de cache très pratique
- ▶ contient de nombreuses fonctions utilitaires (préconditions, chaînes de caractère, etc.)

The logo for Apache Maven, featuring the word "maven" in a bold, lowercase, sans-serif font. The letter 'a' is colored orange, while the remaining letters 'm', 'v', 'e', 'n' are black.



# DIFFICULTÉS RENCONTRÉES

## **Gestion du temps et des imprévus :**

- ▶ algorithme de détection des plans
- ▶ bibliothèque Xuggler

## **Absence de modèles existants :**

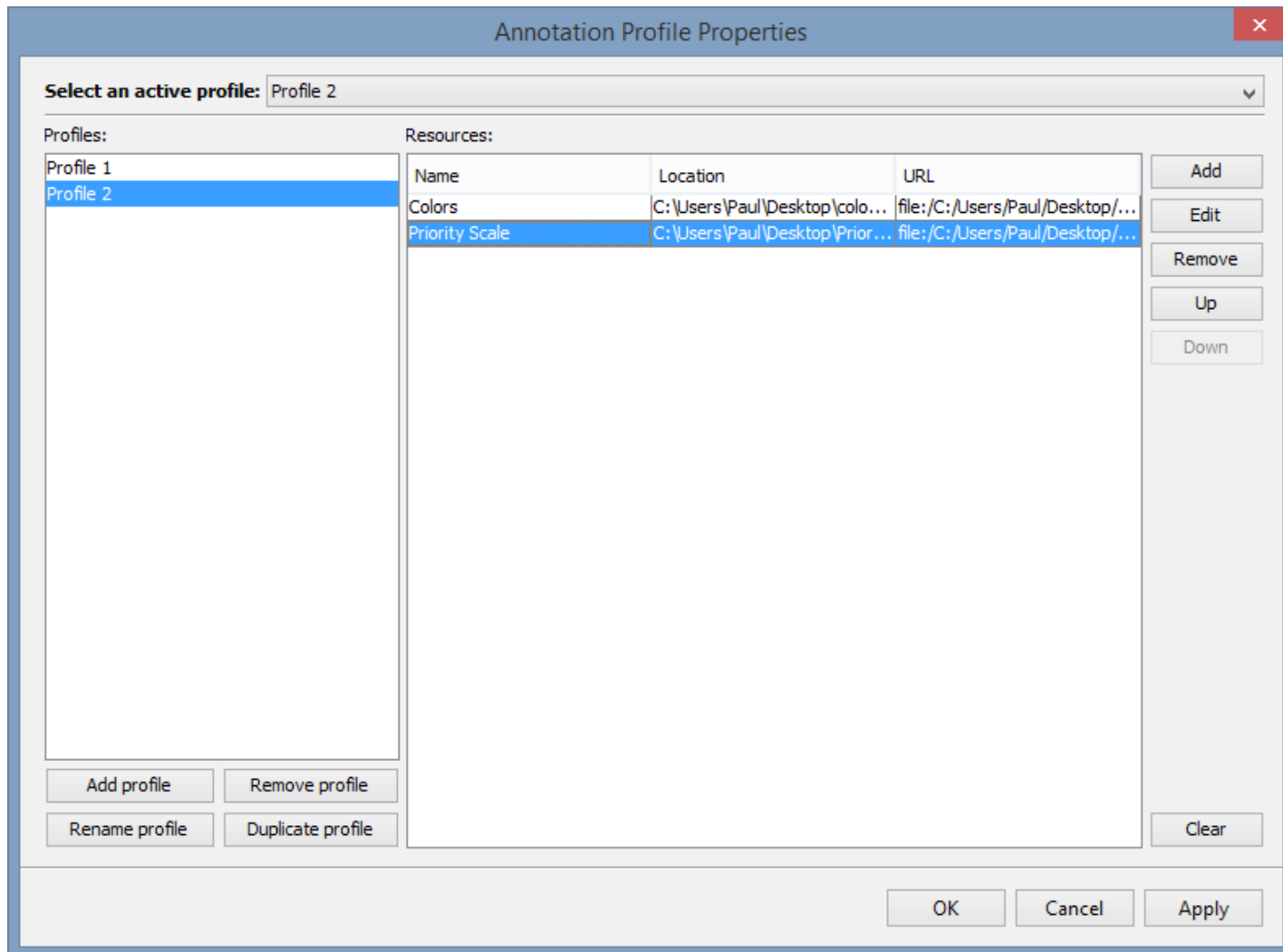
- ▶ contraintes de découplage
- ▶ contraintes d'optimisation

## **MPEG-7 :**

- ▶ document très dense
- ▶ dilemme entre indépendance et respect du standard

# Conclusion

# Gestion des schémas de classification



# Détection automatique des plans

Automatic Shot Boundary Detection ×

Adaptive thresholding parameters Edge-change ratio parameters

Window width:

Percentage of candidates:

The **window width** determines the number of feature vectors to take into account when averaging the neighborhood of each feature. The total number of elements is  $2 * (\text{window width}) + 1$ .

The **percentage of candidates** is the percentile which is used to keep the frame transitions with the highest score.

Reset all parameters to their default values:

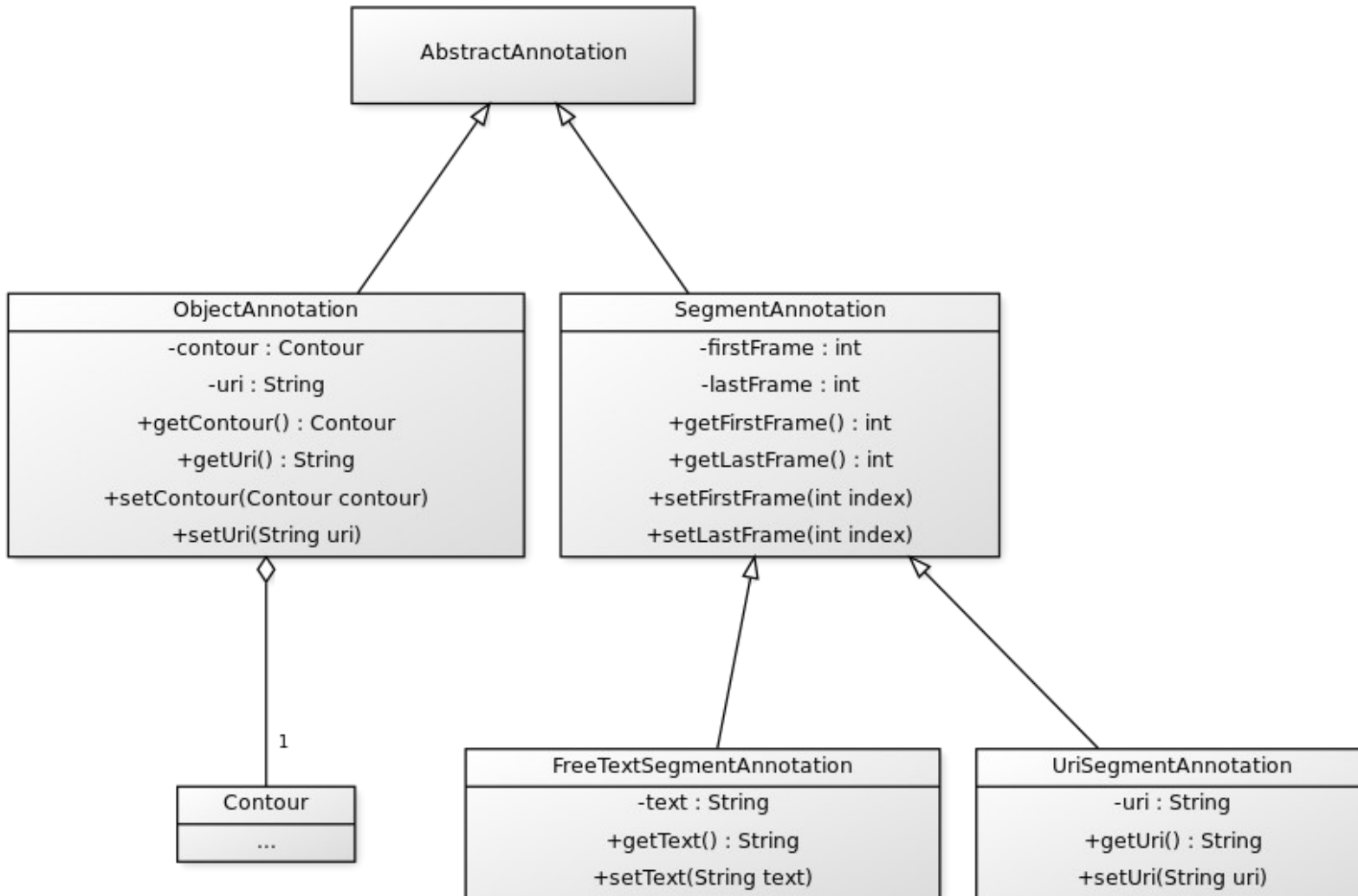
# Schéma de classification en XML (MPEG-7)

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <mpeg7:Mpeg7
3    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4    xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004"
5    xsi:schemaLocation="urn:mpeg:mpeg7:schema:2004 mpeg7-v2.xsd">
6
7    <mpeg7:Description xsi:type="mpeg7:ClassificationSchemeDescriptionType">
8      <mpeg7:ClassificationScheme uri="urn:PriorityScaleCS">
9
10     <mpeg7:Term termID="1">
11       <mpeg7:Name xml:lang="en">0.00</mpeg7:Name>
12       <mpeg7:Definition xml:lang="en">No_Priority</mpeg7:Definition>
13     </mpeg7:Term>
14
15     <mpeg7:Term termID="2">
16       <mpeg7:Name xml:lang="en">0.25</mpeg7:Name>
17       <mpeg7:Definition xml:lang="en">Low_Priority</mpeg7:Definition>
18     </mpeg7:Term>
19
20     <mpeg7:Term termID="3">
21       <mpeg7:Name xml:lang="en">0.50</mpeg7:Name>
22       <mpeg7:Definition xml:lang="en">Middle_Priority</mpeg7:Definition>
23     </mpeg7:Term>
24
25     <mpeg7:Term termID="4">
26       <mpeg7:Name xml:lang="en">0.75</mpeg7:Name>
27       <mpeg7:Definition xml:lang="en">High_Priority</mpeg7:Definition>
28     </mpeg7:Term>
29
30     <mpeg7:Term termID="5">
31       <mpeg7:Name xml:lang="en">1</mpeg7:Name>
32       <mpeg7:Definition xml:lang="en">Top_Priority</mpeg7:Definition>
33     </mpeg7:Term>
34
35     </mpeg7:ClassificationScheme>
36   </mpeg7:Description>
37 </mpeg7:Mpeg7>
```

# Schéma de classification en YAML

```
1 namespace: PriorityScaleCS
2 terms:
3   1:
4     definitions:
5       - lang: en
6         value: No_Priority
7     names:
8       - lang: en
9         value: 0.00
10  2:
11    definitions:
12      - lang: en
13        value: Low_Priority
14    names:
15      - lang: en
16        value: 0.25
17  3:
18    definitions:
19      - lang: en
20        value: Middle_Priority
21    names:
22      - lang: en
23        value: 0.50
24  4:
25    definitions:
26      - lang: en
27        value: High_Priority
28    names:
29      - lang: en
30        value: 0.75
31  5:
32    definitions:
33      - lang: en
34        value: Top_Priority
35    names:
36      - lang: en
37        value: 1
```

# Modélisation des annotations



# Modélisation des contours

