

M1 ALMA
Université de Nantes
2011-2012

Projet de Travaux pratiques :
TOA
Eclipse Vision System Plugin

BIZET Jonathan
CHALIN Jeremy
MARGUERITE Alain
RINCÉ Romain

Université de Nantes
2 rue de la Houssinière, BP92208, F-44322 Nantes cedex 03, FRANCE

Encadrant : Gilles Ardourel, Dalila Goudia, Sagar Sen



UNIVERSITÉ DE NANTES

Table des matières

Table des matières	1
1 Point de vue Utilisateur	2
1.1 Plugins développés	2
1.2 Installation et prise en main	2
2 Point de vue developpeur	3
2.1 Documentation	3
2.2 Conception	4
2.2.1 Architecture	4
2.2.2 Classes	4
2.3 Robustesse	6
2.4 Qualité du code	6
2.5 Facilité de mise en oeuvre des extension	6
Table des figures	7

1 Point de vue Utilisateur

1.1 Plugins développés

Middleware qui est notre plugin principal propose la vue Eclipse suivante : **image manquante**.

Load Video L'utilisateur doit au préalable avoir renseigné le chemin de la vidéo à analyser. Le plugin Acquisition est sollicité au déclenchement de ce bouton.

Lancer Analyse Appel du plugin Reasoning qui effectue une analyse de la vidéo précédemment acquise.

Post L'utilisateur doit au préalable avoir renseigné l'accès token nécessaire pour poster sur une plateforme en ligne. Le plugin en charge de poster sur un blog ou un réseau social est appelé.

1.2 Installation et prise en main

2 Point de vue developpeur

2.1 Documentation

qsdqsd

2.2 Conception

2.2.1 Architecture

Dans cette partie nous détaillerons l'architecture de notre application, à travers ses différents points d'extensions.

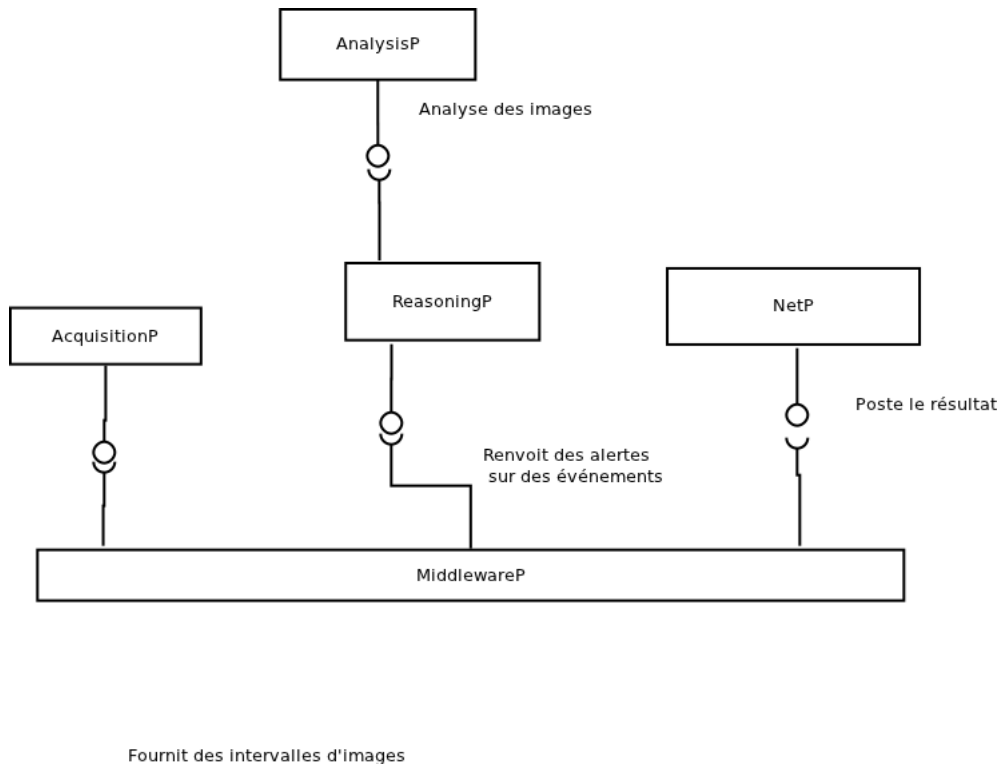


FIGURE 2.1 – Architecture de l'outil

2.2.2 Classes

dsqdqs

Architecture qsdqs

Acquisition Le plugin Acquisition consiste à récupérer un fichier vidéo et en extraire une image périodiquement. Le plugin répond aux spécifications fournies par l'interface IFlux (cf. Fig 2.2).

Pour pouvoir lancer le plugin il est nécessaire d'appeler au minimum les méthodes suivantes dans cet ordre :

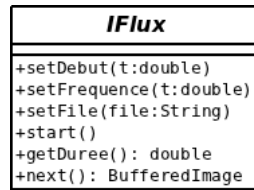


FIGURE 2.2 – Interface IFlux

1. **setFile(file)** pour fournir le fichier vidéo à analyser.
2. **start()** Cette méthode permet de paramétrer l'acquisition et de charger un player.
3. **next()** renvoie un **BufferedImage** et avance dans la vidéo.

Les autres méthodes de l'interface IFlux sont :

- **setDebut(tEnSecondes)** pour donner l'instant à partir duquel la vidéo sera analysée. Cette méthode doit être lancée avant **start()** ou alors **start** est relancée
- **setFrequence(tEnSecondes)** Le nom de la méthode est trompeuse mais elle permet de donner la période en secondes à laquelle le plugin récupère un image dans la vidéo. La période peut être changée avant ou en cours d'acquisition. Cependant le pointeur sur la prochaine image à acquérir étant déjà défini, le changement de période ne sera effectif qu'à partir du second **next()**
- **getDurée()** renvoie un le temps en secondes de la vidéo. **start()** doit avoir été déjà lancée.

Pour effectuer l'acquisition le plugin utilise la librairie JMF qui est malheureusement peu pratique puisqu'elle ne gère qu'un faible nombre de formats vidéo. De plus nous avons pu constater un bug au niveau de la méthode qui renvoie la durée de la vidéo **Player.duration()** ; en effet celle-ci renvoie une valeur bien trop grande par rapport à la longueur réelle de la vidéo.

Reasoning sdqsd

NetP Le NetP est dédié à poster des informations sur un réseau social ou un blog. Un contributeur de ce point d'extension doit être en mesure de proposer une solution pour chaque action que requière Middleware. Ces actions sont données dans l'interface **IInformation** suivante :

Le contributeur que nous avons implémenté propose d'utiliser le réseau social Facebook et utilise la librairie suivante [?].

<i>Information</i>
<code>+postVideo(evFB:IEvent): void</code> <code>+postMessage(evFB:IEvent): void</code> <code>+postVideo(evFB:IEvent): void</code> <code>+postPicture(evFB:IEvent): void</code> <code>+setAccessToken(accesToken:String): voi</code>

FIGURE 2.3 – Interface pour NetP

2.3 Robustesse

sqdqs

2.4 Qualité du code

dqsdqs

2.5 Facilité de mise en oeuvre des extension

Table des figures

2.1	Architecture de l'outil	4
2.2	Interface IFlux	5
2.3	Interface pour NetP	6