

Eclipse Vision System Plugin

Diallo Algassimou

30 mars 2012

Table des matières

1	Introduction	1
2	Architecture	2
3	bibliothèque requise	3
4	PagePreference	3
5	exécution	3
5.1	imageAcquisitionVideo	4
5.2	imageAnalysis	4
5.3	imageReasoning	4
5.4	imagePublishBlogger	6
6	Points d'extension et interfaces	6
6.1	acquisition d'image	6
6.2	analyse d'image	7
6.3	raisonnement	8
6.4	publication de post	8

Présentation générale

1 Introduction

Eclipse Vision Système permet de détecter des faits à partir d'un flux d'images et de créer des posts sur un réseau social. L'origine du flux d'images,

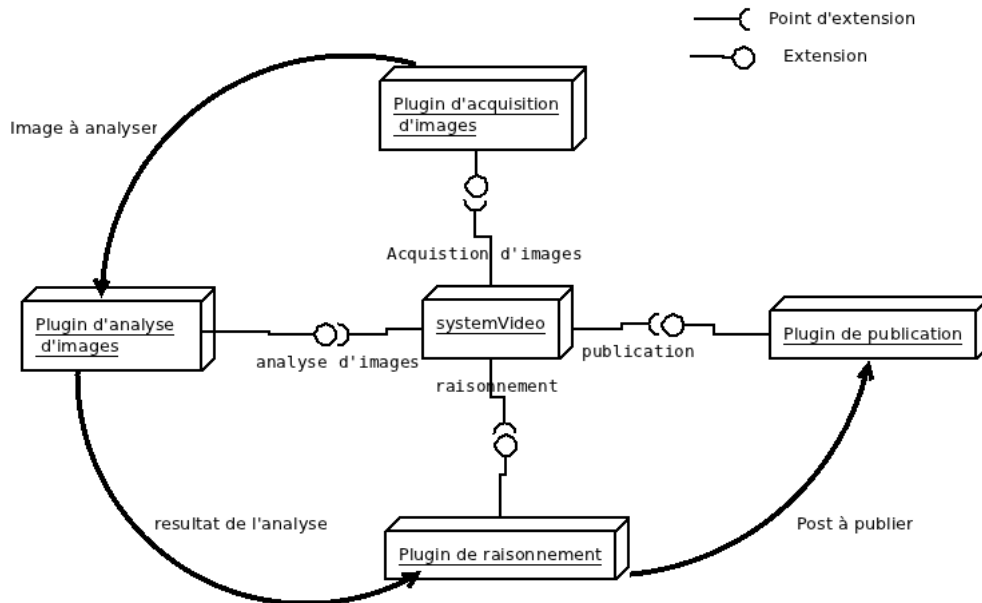


FIGURE 1 – Architecture

les traitements apportés aux images ainsi que les faits recherchés dans celles-ci et les posts publiés sur un réseau social sont autant de points de variations qui sont pris en compte par Eclipse Vision Système.

2 Architecture

Eclipse Vision Système définit deux plugins pour des points d'extensions d'eclipse :

- actionSets (pour lancer le système) (plugin systemVideo).
- PreferencePages (Permet d'avoir une page de préférences pour notre système) (plugin PreferencePages).

systemVideo est le plugin centrale de notre système. Il fournit quatre points d'extensions qui permettent de capturer toutes les variations possibles dans l'utilisation du système. Ces points d'extensions sont les suivants :

- *imageAcquisitionExt* : point d'extension pour les plugins d'acquisition (capture) d'images ;
- *imageAnalysisExt* : point d'extension pour les plugins d'analyse d'images ;
- *imageReasonigExt* : point d'extension pour les plugins de traitement d'images ou de raisonnement avant publication ;
- *imagePublicationExt* : point d'extension pour les plugins de publication de posts.

Les plugins suivants ont été développés et peuvent être utilisés :

- **imageAcquisitionCamera** : plugin de capture d'une image à partir d'une webcam ;
- **imageAcquisitionVideo** : plugin de capture d'une image à partir d'une vidéo ;
- **imageAnalysis** : plugin d'analyse d'une image (reconnaît des visages dans une image) ;
- **imageReasoningSimple** : Publie si l'image contient plus d'un certains nombre de visages ;
- **imagePublishBlogger** : publie Un message sur un google blog.

Installation

3 bibliothèque requise

les bibliothèques suivantes sont requises pour pouvoir faire fonctionner notre système :

- OpenCV
- javaCV
- xuggler

Nos plugins ne fonctionnent qu'en les lançant avec eclipse.

Tutoriel

4 PagePreference

La page de préférences permet pour chacun des points d'extensions à l'utilisateur de choisir le plugin qu'il veut parmi tous ceux disponibles, et aussi il peut renseigner l'adresse d'un serveur proxy et le port de celui-ci. On arrive à la page de préférences en cliquant sur : window > Preferences > systemVideo.

5 exécution

Dans cette exécution nous allons considérer que les options ont été fixées comme le montre la figure 2.

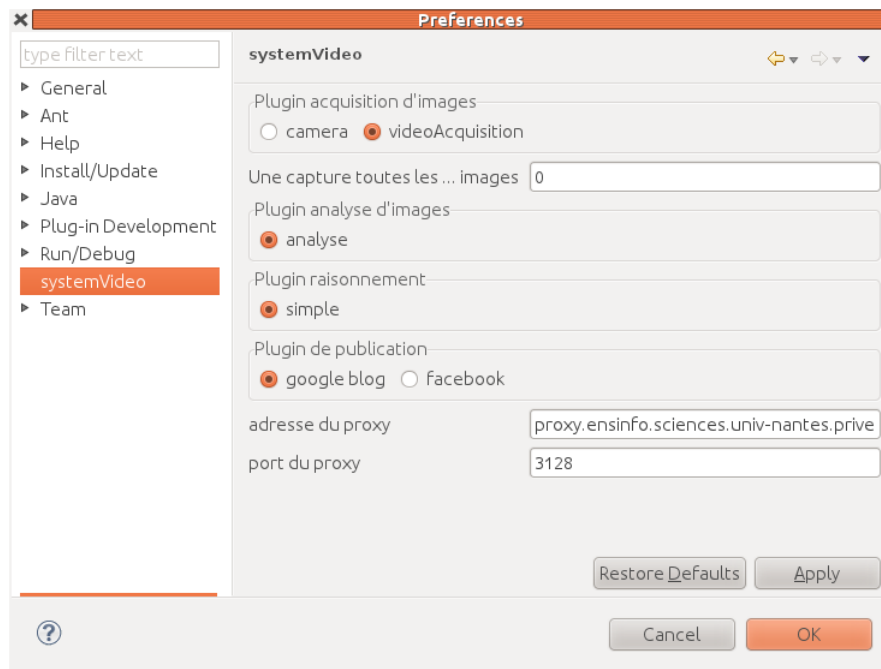


FIGURE 2 – Page de préférences

5.1 imageAcquisitionVideo

Ce plugin va capturer les images partir d'une vidéo pour cela il va demander le chemin de la vidéo (figure 3).

5.2 imageAnalysis

Ce plugin analyse une image à la recherche de visages. Pour cela il a besoin d'un fichier permettant de définir ce qu'est un visage. Ce fichier est fourni par défaut avec les sources de openCV il suffit de renseigner le chemin du fichier (haarcascade_frontalface_alt.xml) dans la fenêtre qui s'ouvre (voir figure 5).

5.3 imageReasoning

Ce plugin compte le nombre d'images trouvées par le plugin précédent et post un billet sur le blog si le nombre de visage dépasse un certains nombre (Ce nombre est renseigné par l'utilisateur comme le montre la figure ??).

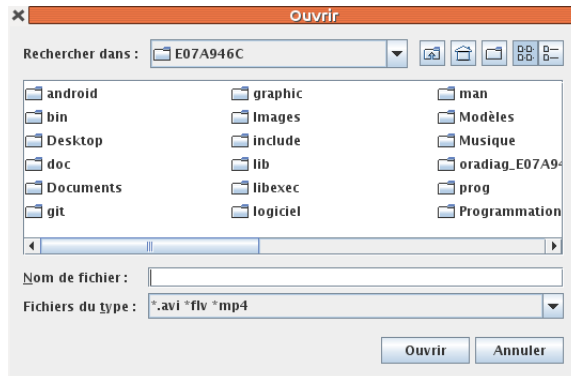


FIGURE 3 – Chemin du fichier vidéo

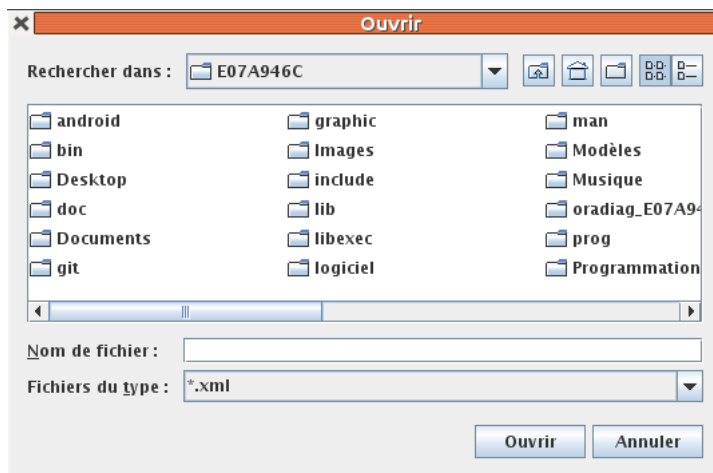


FIGURE 4 – fichier définissant un visage

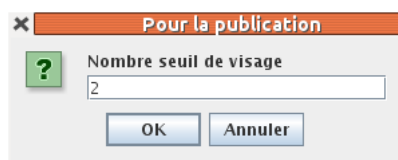


FIGURE 5 – Le nombre seuil de visages

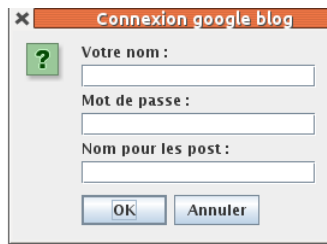


FIGURE 6 – identifiant de connexion compte google

5.4 imagePublishBlogger

Ce plugin permet la publication de post sur un blog google. Pour cela il demande en paramètres les identifiant de connexion ainsi le nom de l’auteur du post (voir figure 6).

Utilisation avancée

Comme le montre la figure 1 systemVide définit quatre points d’extensions. Pour écrire un plugin pour systemVideo on choisit l’un de ces points d’extensions et on définit une classe qui implémente l’interface du point d’extension.

Notre système est un chaînage de plugin car le plugin d’acquisition d’image capture une image et la passe au plugin d’analyse qui lui analyse l’image et passe ces résultats au plugin de raisonnement qui traite les résultats et décide de publier ou non.

6 Points d’extension et interfaces

6.1 acquisition d’image

L’interface de ce point d’extension est la suivante :

```
package interfaces ;

public interface IImageAcquisition {
    public void run () ;
    public void init () ;
    public void setIImageAnalysis (IImageAnalysis analyse) ;
}
```

méthode run

Cette méthode est appelé par le plugin centrale (systemVideo). elle lance réellement l'exécution du système.

méthode init

Cette méthode est appelé pour initialiser le plugin. Elle est appelé avant la méthode run et peut être utilisé pour demander des informations à l'utilisateur (chemin d'un fichier , un identifiant etc).

méthode setImageAnalysis

Cette méthode permet de fixer le l'objet qui va analyser les images capturées.

6.2 analyse d'image

```
package interfaces ;

import java .awt .image .BufferedImage ;

public interface IImageAnalysis {
    public void analyse (BufferedImage img);
    public void init ();
    public void setImageReasoning (IImageReasoning imgR);
}
```

méthode analyse

Cette méthode est appelé par le plugin d'acquisition d'image et à comme paramètre l'image capturer.

méthode init

Cette méthode est appelé une seule fois (avant toute autres méthodes) et peut permettre au plugin de s'initialiser.

méthode setImageReasoning

Cette méthode permet de fixer le l'objet qui va recevoir les résultats de nos analyse.

6.3 raisonnement

```
package interfaces ;

import java.util.List ;

public interface IImageReasoning {
    public void reasonnig (List<Object> o) ;
    void init () ;
    public void addIImagePublish (IImagePublish imgP);
}
```

méthode reasoning

Cette méthode est appelé par le plugin d'analyse d'image et à comme paramètre une liste d'objets issue de l'analyse. On peut remarquer que cette liste peut être une liste de n'importe qu'elle objet java, ainsi pour pouvoir raisonner au mieu ce plugins doit connaître qu'elle sont les type concrets de objets envoyer et dans qu'elle ordre ce qui crée une forte dépendances entre le plugin d'analyse et le plugin de raisonnement.

méthode init

Cette méthode est appelé une seule fois (avant toute autres méthodes) et peut permettre au plugin de s'iniatialiser.

méthode addIImagePublish

Cette méthode permet d'ajouter un IImagePublis (un plugin pour la publication de post ceux connu du plugin de raisonnement). Ainsi un même peut être publier sur différents réseaux sociaux.

6.4 publication de post

```
package interfaces ;

public interface IImagePublish {
    void publish (String title , String content) ;
    void init () ;
    void setProxyHost (String proxyHost);
    void setProxyPort (String proxyPort);
}
```


méthode publish

Cette méthode est appelé par le plugin de raisonnement. Elle prend en paramètre l'image à publier, le titre du post et le contenu du post.

méthode init

Cette méthode est appelé une seule fois (avant toute autres méthodes) et peut permettre au plugin de s'iniatialiser.

méthode setProxyHost

Cette méthode permet de prendre en compte l'adresse du proxy. Elle est appelé une seule fois et avant la méthode publish

méthode setProxyPort

Cette méthode permet de prendre en compte le port du proxy. Elle est appelé une seule fois et avant la méthode publish