RETRO 2014 Developper's Guide

- Development Tools
- Architecture
- Clustering method & Descriptor

How to Contact Us:

Jean-Yves RAMEL (ramel@univ-tours.fr)
Frédéric RAYAR (rayar@univ-tours.fr)

Laboratoire d'Informatique
Equipe Reconnaissance des formes et analyse d'images
64, avenue Jean Portalis
37200 – Tours
France

For more contact information, please refer to the PaRADIIT project website
https://sites.google.com/site/paradiitproject/

Licence

RETRO 2012
Copyright © RFAI, LI Tours, 2011-2012

Partners

The authors would like to thank all the members of the BVH-CESR for their collaboration. This work has been supported by the Google Digital Humanities research Awards given to the Computer Science Laboratory of Tours (RFAI team).

UNIVERSITÉ FRANÇOIS - RABELAIS TOURS	http://international.univ-tours.fr/
POLYTECH TOURS	http://polytech.univ-tours.fr/
LI Laboratoire d'Informatique EA 2101	http://www.li.univ-tours.fr/
RFAI Reconnaissance des formes LI Analyse d'images	http://www.rfai.li.univ-tours.fr/
Google™	http://www.google.fr/intl/en/about/
CESR Centre d'Étude Supérieure de la Renaissance	http://cesr.univ-tours.fr/
BVH	http://www.bvh.univ-tours.fr/

Content

PART I

DEVELOPMENT TOOLS

Development constraints

- **IDE** : Visual Studio 2012

- **Language** : C# with .NET 4.5 Framework

If the developer wants to modify the RETRO source code he has to develop in **C#** and use at least the **4.5 version of .NET** framework and the **2012 version of Visual Studio**.
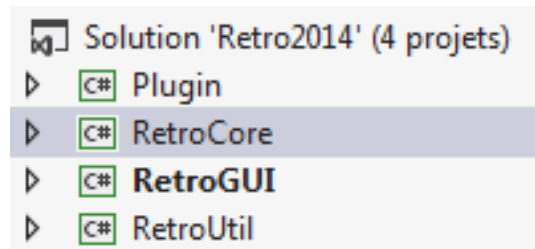
In addition, RETRO has been developed according the **MVVM** (Model – View – ViewModel) design pattern. So it's important the developer respect this model in developing RETRO.

PART II
SOLUTION ARCHITECTURE

Projects

The RETRO solution comprised 4 projects :

1. **Plugin**	This project contains the database object model. It contains the tools to read and use a clustering method (except the Template Matching method which is defined in the **Clustering** project), a descriptor or a document reader plugin. Normally, the developer don't have to modify it.
2. **RetroCore**	This project is the principal project of the solution. It permit also to define a RETRO model project and the returned values. It contains the clusters and shapes characteristics and all the Template matching clustering method features.
3. **RetroGUI**	This project contains the principals' views of the project and the principal methods too.
4. **RetroUtil**	This project contains various small tools, like dynamic splash for example.

Package diagram

Here is the package diagram of RETRO:
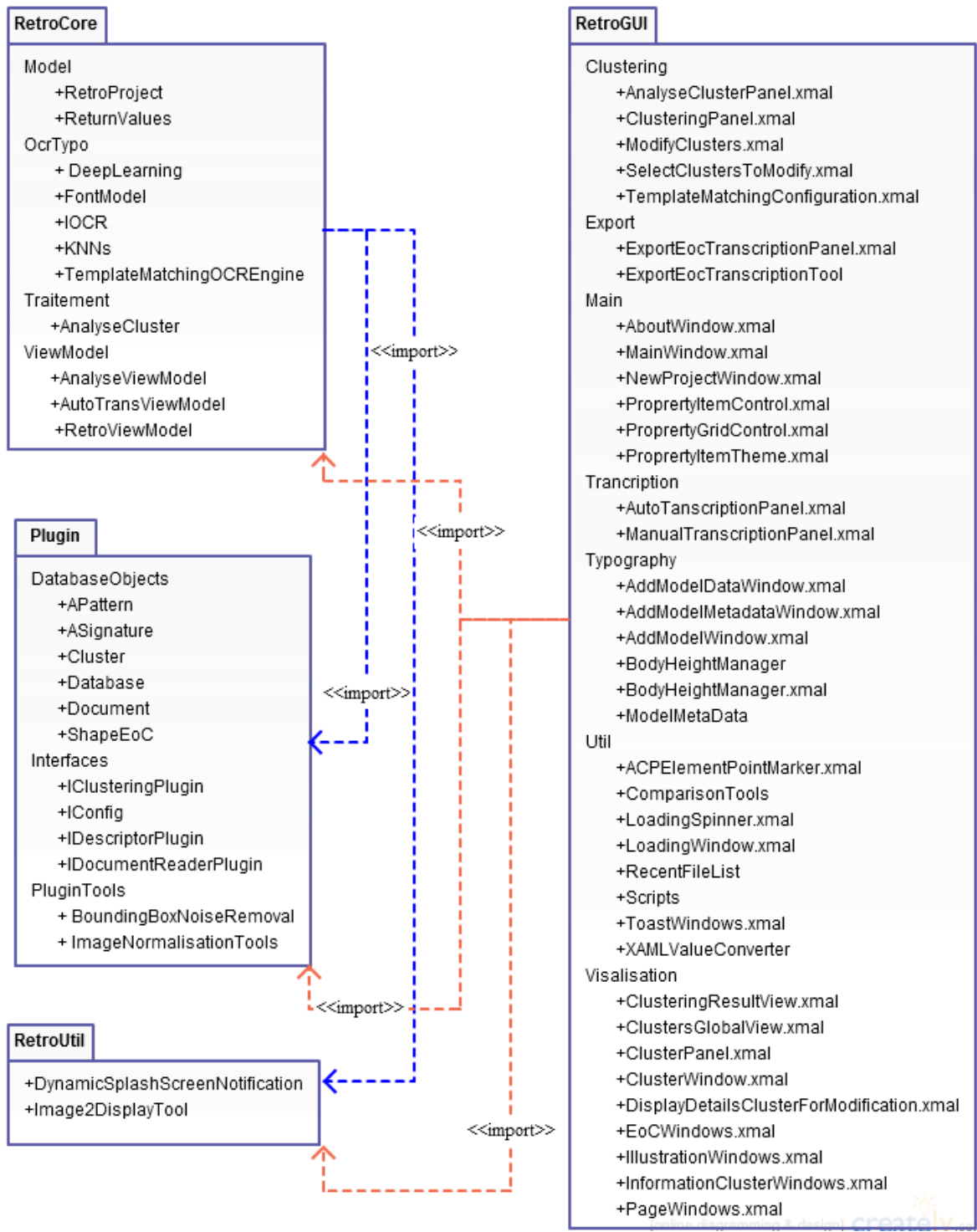
PART III

CLUSTERING METHODS & DESCRIPTORS

Plugins already implemented

RETRO contains already many clustering methods and descriptors in order to realize a clustering.

The plugins already implemented are in the directory:
ExternalLib\DLLs_MethodesClustering_Descripteurs

- Descripteurs
- DocumentReader
- MethodesClustering
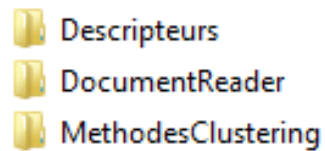
Each directory contains respectively:

- **Descriptors**: DirectionalDescriptorPlugin and ZernikeDescriptorPlugin
- **Document Reader**: AltoReaderPlugin
- **Clustering methods**: BIRCHClusteringPlugin and KMedoidClusteringPlugin plugins.

Develop a clustering method

If the developer wants to develop a new clustering method, he has to develop it according the *ICLusteringPlugin* Interface in the project **Plugin**.
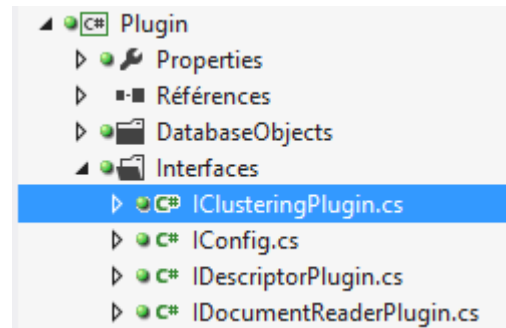

This interface contains all the method prototypes that the developer has to develop to create a new clustering method.

Develop a descriptor

To develop a descriptor, the developer has to do the same thing that for a clustering method, except he has to follow the *IDescriptorPlugin* Interface to develop his own descriptor.

Develop a document reader

Actually RETRO permit to read only one type of files to realize the clustering: File ALTO + Pictures and a document reader for this file is already implemented.

However, if the developer wants to develop his own reader document, he had to consult the *IDocumentReader* Interface.

Add a clustering method

If the developer want to add his clustering method (which he has developed like it is explained in the "Develop a clustering method" part) he has to follow 2 steps:

1. Add his plugin (*.dll) in the right directory:
   ExternalLib\DLLs_MethodesClustering_Descripteurs\MethodClustering

2. Complete the Xml file **ClusteringMethods** presents in the **XML_Files** director according the given model :

```
<Method>
      <Name>Name of the clustering method to add</Name>
      <Description>Write here the description of the clustering
      method to add.
      </Description>
      <Path>
      ..\\..\\..\\ExternalLib\\DLLs_MethodesClustering_Descripteur
      s\\MethodesClustering\\PluginOfTheClusteringMethodToAdd.dll
      </Path>
</Method>
```

Then, the clustering method will be automatically added to RETRO and will be available for the user to realize a clustering.

Add a descriptor

If the developer want to add his clustering method (which he has developed like it's explain in the "Develop a clustering method" part) he has to follow 2 steps:

1. Add his plugin (*.dll) in the right directory:
   ExternalLib\DLLs_MethodesClustering_Descripteurs\Descripteurs

2. Complete the Xml file *Descriptors* presents in the *XML_Files* director according the given model :

```
<Descriptor>
      <Name>Name of the descriptor to add</Name>
      <Description>Write here the description of the descriptor to
add.
      </Description>
      <Path>
      ..\\..\\..\\ExternalLib\\DLLs_MethodesClustering_Descripteur
      s\\MethodesClustering\\PluginOfTheDescriptorToAdd.dll
      </Path>
</Descriptor>
```