

PIROser - Parseur UML du Diro

Truong Pham(PHAL29018809)

2 avril 2013

Résumé

Parseur de diagramme de classes UML fait avec amour.

1 Introduction

La deuxième itération de PIROser consiste à ajouter une fonctionnalité permettant le calcul de certaines métriques sur les classes parsées auparavant. Aussi, à la demande de l'utilisateur, PIROser peut générer un fichier CSV contenant le nom des classes avec leurs métriques respectives.

2 Conception

L'extension de PIROser se consiste entièrement de la nouvelle classe **Metric**. La modularité de la première version de l'application m'a permise de rajouter des fonctions sans toucher aux anciennes composantes. Grâce à cela, j'ai pu étendre le logiciel sans avoir peur de "briser" ce qui a été fait auparavant et, en plus, mon temps de développement a été raccourci car je n'ai pas eut à travailler sur mon ancien code.

Afin d'initialiser le calcul des métriques, il faut créer un nouvel objet **Metric** et passer comme arguments le **Model** déjà parsé ainsi que le nom de la **Classe** que l'on veut "questionner". Tous les calculs des différentes métriques sont définis par les fonctions portant leurs acronymes respectifs. Par exemple, `get_ana()` calculera le **Average Number of Arguments**. Ces calculs sont exécutés à chaque fois que l'on sélectionne une **Classe**.

Il peut sembler bizarre que la majorité des fonctions calculant les métriques sont définies deux fois, une fois sans paramètre et l'autre avec. La première méthode, sans argument, sert d'interface publique pour les autres classes du programme tandis que la seconde méthode est privée et permet le calcul des métriques qui invoque le principe "d'héritage". Ce choix permet un calcul récursif et facilite, selon moi, la compréhension du code versus un calcul itératif contenant tellement de boucles que la fonction ressemble à une tornade.

Les détails pointus de chaque méthode sont expliqués dans les commentaires de mon code. Je doute que tu n'aies envie de lire le pourquoi de chaque boucle de mon code dans mon rapport et je n'ai que 3 pages.

3 Amélioration possible

Pour un diagramme de classe UML léger, comme celui fourni pour le travail, le calcul de tous les métriques est instantané, du moins sur ma machine. Cependant, si le diagramme était vraiment lourd et que le calcul nécessite quelques secondes pour chaque **Classe**, cela pourrait rendre l'expérience de l'utilisateur aussi terrible que d'utiliser Eclipse dans les laboratoires du DIRO. Pour remédier à cette situation, une méthode possible serait de calculer tous les métriques de chaque classe immédiatement après le parsing du diagramme et de mettre en "cache" les résultats. Le calcul ne se ferait qu'une seule fois, donc un seul délai où on pourrait afficher un beau "splash screen" du DIRO. L'expérience dans le "GUF" serait beaucoup plus agréable et on entendrait moins notre processeur crier.

4 Exécution

Pour exécuter le programme :

```
include jdk-1.7
ant run
```

Pour exécuter les tests :

```
ant test
```

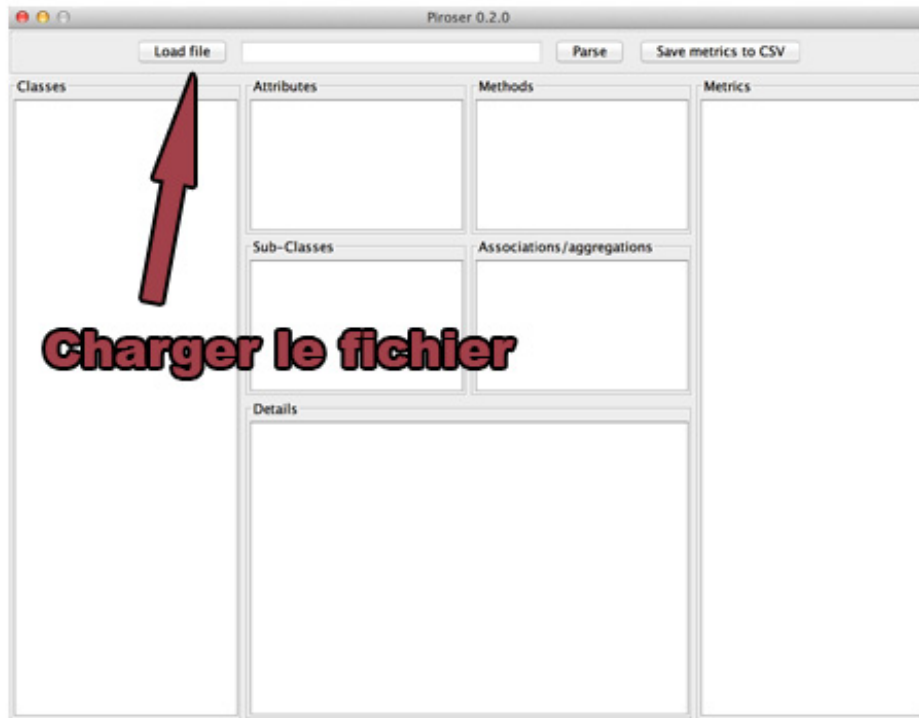


FIGURE 1 – Charger un fichier

5 Utilisation

5.1 Parsing

Pour débiter, il faut charger le fichier comme à la figure 1 et cliquer sur le bouton **Parse**. Si le parsing se complète avec succès, les classes seront listées dans la colonne de gauche tel que présenté à la figure 2. Ensuite, lorsque l'on sélectionne chaque classe, les composantes de ces classes et les métriques se calculeront et s'afficheront comme à la figure 3.

5.2 Visualisation des définitions

Pour visualiser les définitions des acronymes de chaque métrique, il suffi de la sélectionner tel que vu à la figure 4.



FIGURE 2 – Parser le fichier

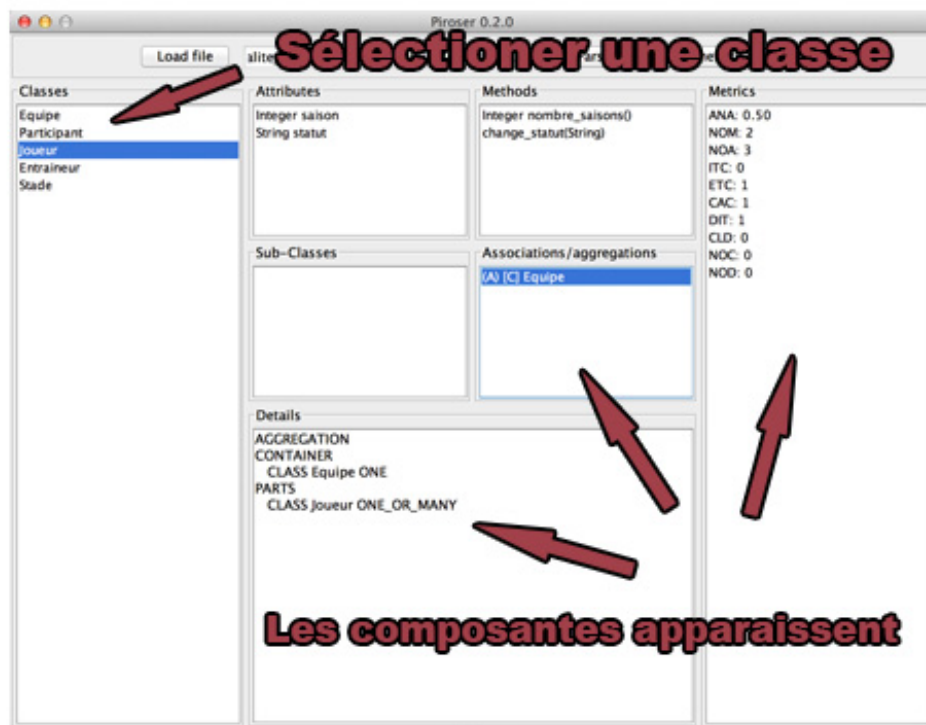


FIGURE 3 – Visualiser les composantes

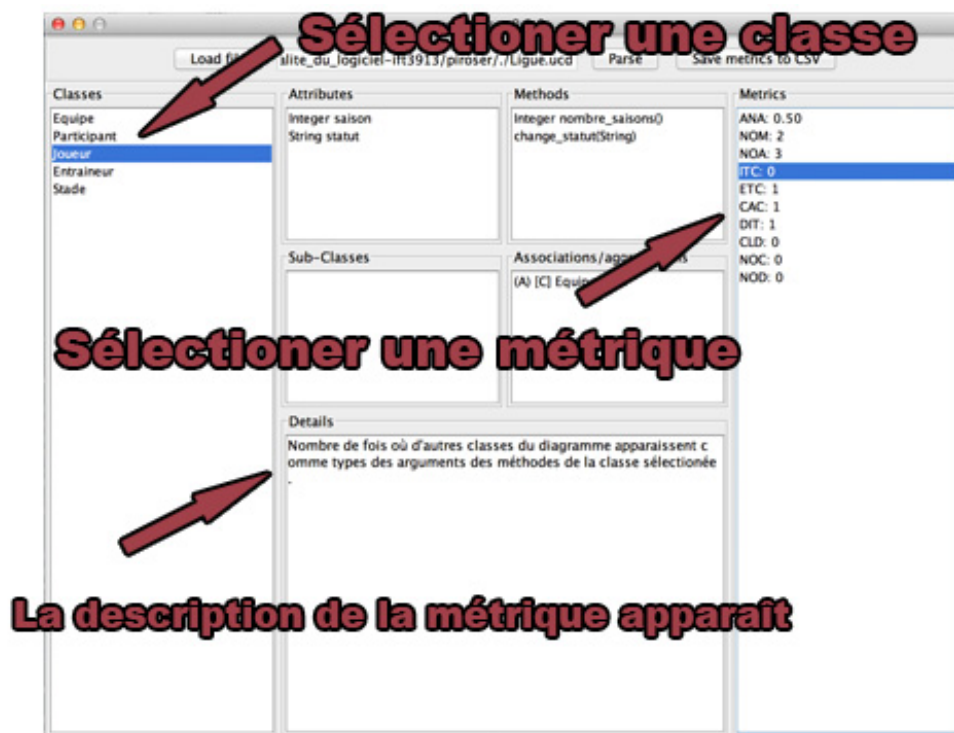


FIGURE 4 – Définitions des métriques

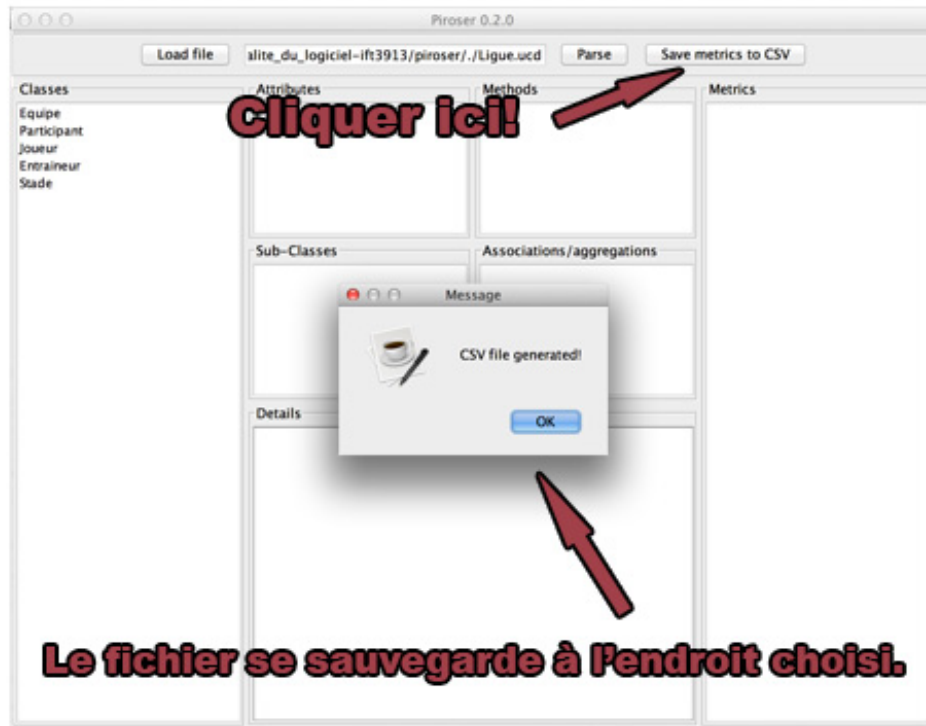


FIGURE 5 – Sauvegarde du fichier CSV

5.3 Génération du CSV

La génération du CSV se fait en cliquant sur le bouton `Save metrics to CSV`. Il faudra choisir un emplacement et un nom de fichier. Si le fichier ne se termine pas par `.csv`, le programme imposera cette extension. Un message de complétion s'affichera s'il n'y a pas eut d'erreurs tel qu'à la figure 5.