

TP1-IFT3913 HIVER 2014
- Rapport -

Tristan Savaria 1062837 (savt09129202) IBrahima Faye 959315 (fayi29129204)

9 février 2014

Table des matières

1	Introduction	3
2	Métrique Implémentées	3
2.1	Métriques Absentes	3
3	Détails et Erreurs	3
4	Commentaire	4

1 Introduction

Dans ce travail pratique, nous avons pour but d'extraire un ensemble de mesures de code Java à partir d'un Workspace. Pour cela des squelettes de programmes nous ont été fournis pour nous faciliter la tâche. Pour notre part nous avons aussi implémenté une nouvelle classe nommée Metrique sur laquelle on peut obtenir toutes les mesures extraites en un seul string sous la forme exigée pour ce projet. En outre nous avons effectué plusieurs affichages sur la console pour la tracabilité. La quasi-totalité des mesures réussies y ont été affichées. Cependant nous n'avons pas réussi à obtenir l'intégral des métriques demandées dans la feuille d'instruction.

2 Métrique Implémentées

Classe : Nom qualifié de la classe (incluant le nom de package)
LOC : Nombre de lignes
NbreInterfaces : Nombre Interfaces de la classes (toutes, transitivement héritées)
NbreAttrImpl : Nombre d'attributs implantés dans la classe
PctAttrPub : Pourcentage des attributs qui sont publiques
PctAttrHer : Pourcentage des attributs hérités
NbreMethImpl : Nombre de méthodes implantées
PctMethPub : Pourcentage des méthodes qui sont publiques
PctMethHer : Pourcentage des méthodes qui sont héritées
DIT : Profondeur dans l'arbre d'héritage
NOC : Nombre de fils
WMC : Somme des complexités des méthodes. Complexité est la complexité cyclomatique.
fanOut : Nombre de classes sont utilisées par la classe

2.1 Métriques Absentes

LCOM : Manque de cohésion
CBO : Couplage entre les objets
fanIn : Nombre de classes qui utilisent la classe

3 Détails et Erreurs

La majorité des métriques extraites ce passent de commentaires. LOC échoue dans certains cas, mais en majorité il ce comporte comme prévue. Dans les classes imbriquées, le nom de la classe et le LOC sera faux. Le nom s'avère être le nom de la classe supérieur et le LOC, le nombre de lignes de la classe imbriquée. Également, dans certain cas LOC est -1 puisque nous utilisons `ASTNode.getStartPosition()` et `ASTNode.getLength()` dans

le calcul et si aucune information de la source de la position est enregistré, le résultat sera -1. Nous n'avons pas réussi à déterminer la raison derrière ce comportement.

Le nombre d'interfaces transitivement hérités comprend des interfaces de librairies java, mais ne comprend pas ceux du type Object.

Les constantes ne sont pas pris en compte pour les attributs implémentés de la classe, ni pour le pourcentage des attributs publiques.

La profondeur dans l'arbre d'héritage cesse avant d'atteindre la classe Object.

Pour le WMC, la complexité cyclomatique total a été implémenté. C'est à dire, le WMC d'une classe est équivalent à la somme du WMC de chaque méthodes de la classe.

fanOut a plusieurs erreurs. Il y a une mauvaise filtration de données. Pour le calcul, nous observons le type des attributs de la classe, des variables déclarées dans les méthodes et le type des variables passées en paramètre dans les méthodes. Pour les attributs d'une classe et ceux déclarés dans le méthodes, les types primitifs sont exclus. Cependant, il aurait été mieux d'exclure le type String et les collection java. Pour les paramètres des méthodes, seulement int, String et char sont exclus. Si un tableau de ces types est utilisé, il sera calculé. Cette mauvaise filtration nous donne une mesure de fanOut innapproprié quoique prévisible si l'on connait l'implémentation.

4 Commentaire

LCOM, CBO et fanIn sont des mesures nécessitant d'établir une relation entre deux classes. Avec l'Abstract Syntax Tree extrait, il est facile d'établir une relation de parenté ou de voir si notre classe courante (qui est en train d'être visitée) utilise un type étranger à elle-même. Il n'est pas aussi facile d'établir la relation dans les deux sens puisqu'il faut revisiter l'arbre en entier tout en ayant en mémoire notre visite précédente pour pouvoir en tirer des conclusions.

Ces mesures, ainsi que fanIn et NOC ont été nos difficultés majeures dans ce travail. Une mauvaise connaissance de l'Abstract Syntax Tree et de ses noeuds nous a considérablement ralenti dans notre progression.