



Mesures de maintenabilité de logiciels

Projet de
Programmation



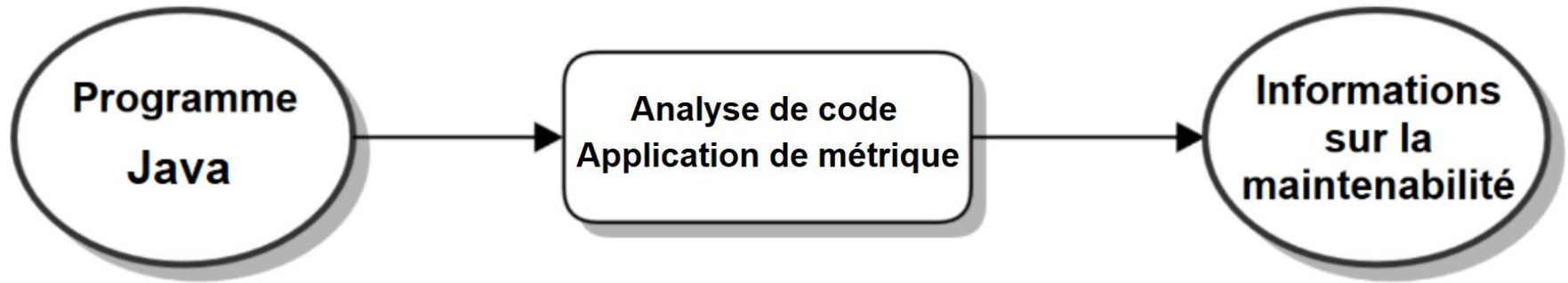
Delrée Sylvain
Giachino Nicolas
Martinez Eudes
Ousseny Irfaane

Introduction

Métrique de Martin

Analyse de dépendances

Étude de Maintenabilité



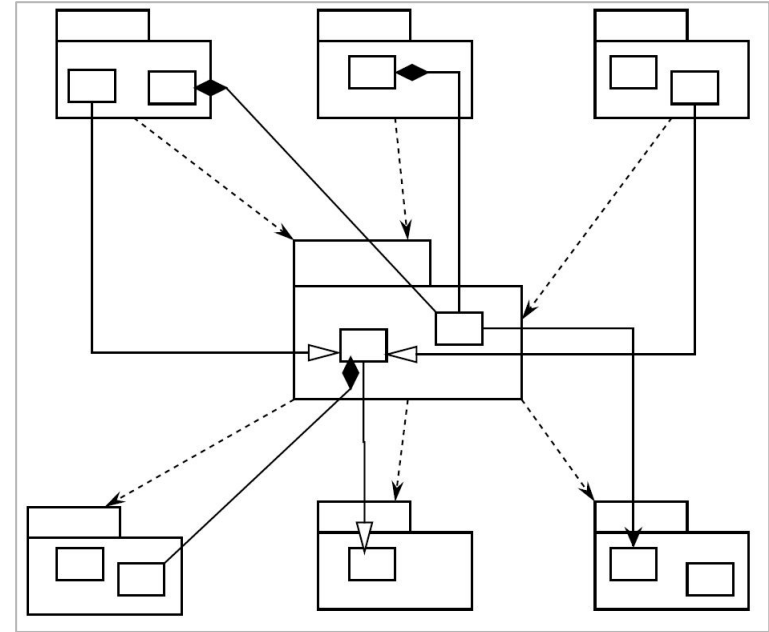
Domaine - (1) Design & Maintenabilité

Granularités :

- Attributs et méthodes
- Classes et objets
- Packages, Namespaces
- Modules

Dépendances :

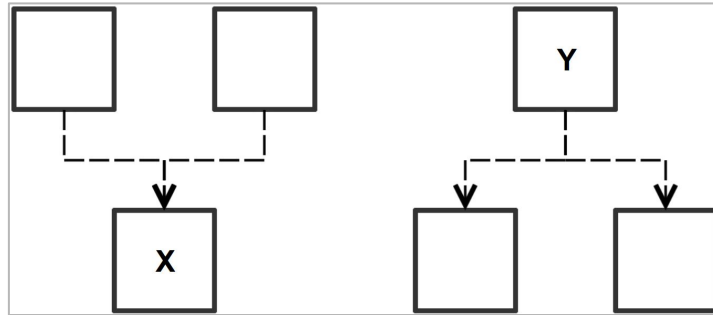
- Dépendance par héritage
- Dépendance par association
- Dépendance par utilisation



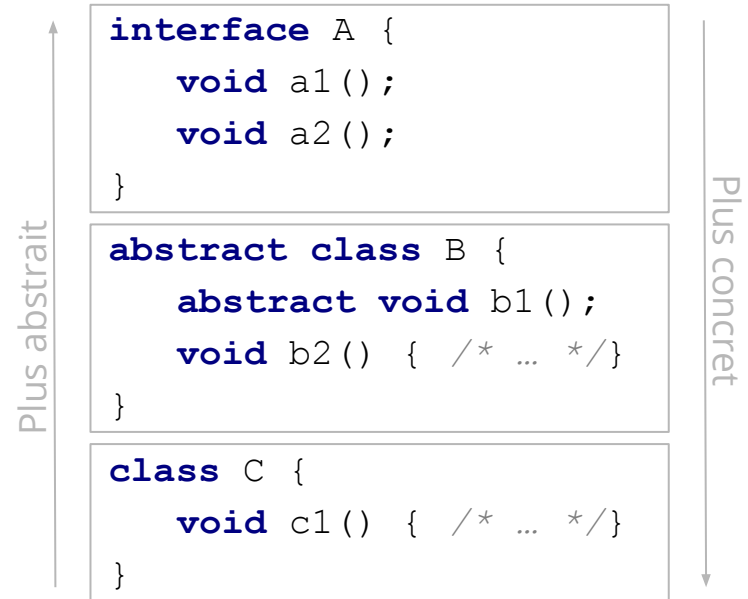
Domaine - (2) Stabilité et Abstraction

Stabilité

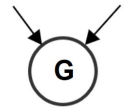
Déterminée depuis le couplage
Responsabilité & Indépendance



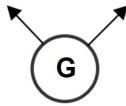
Abstraction



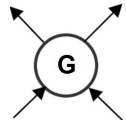
Domaine - (3) Métrique de Martin



- Couplage afférent (**Ca**)



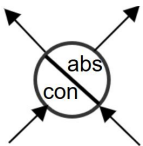
- Couplage efférent (**Ce**)



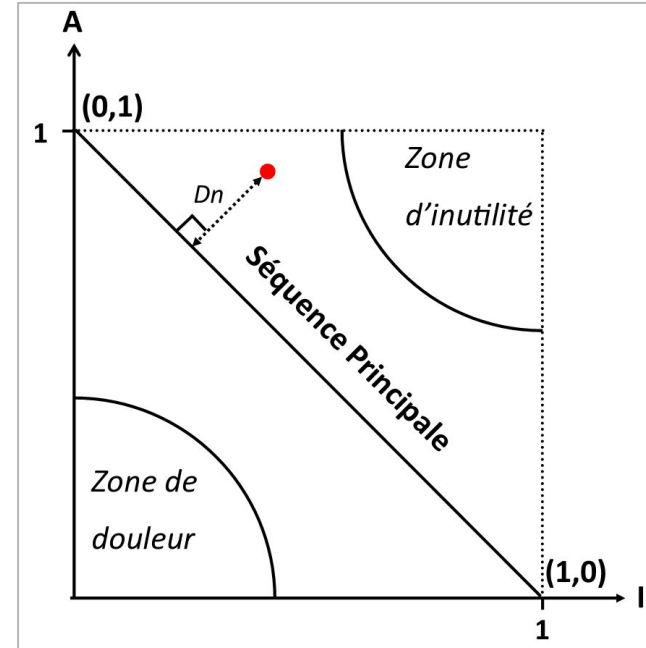
- Instabilité (**I**)



- Niveau d'abstraction (**A**)



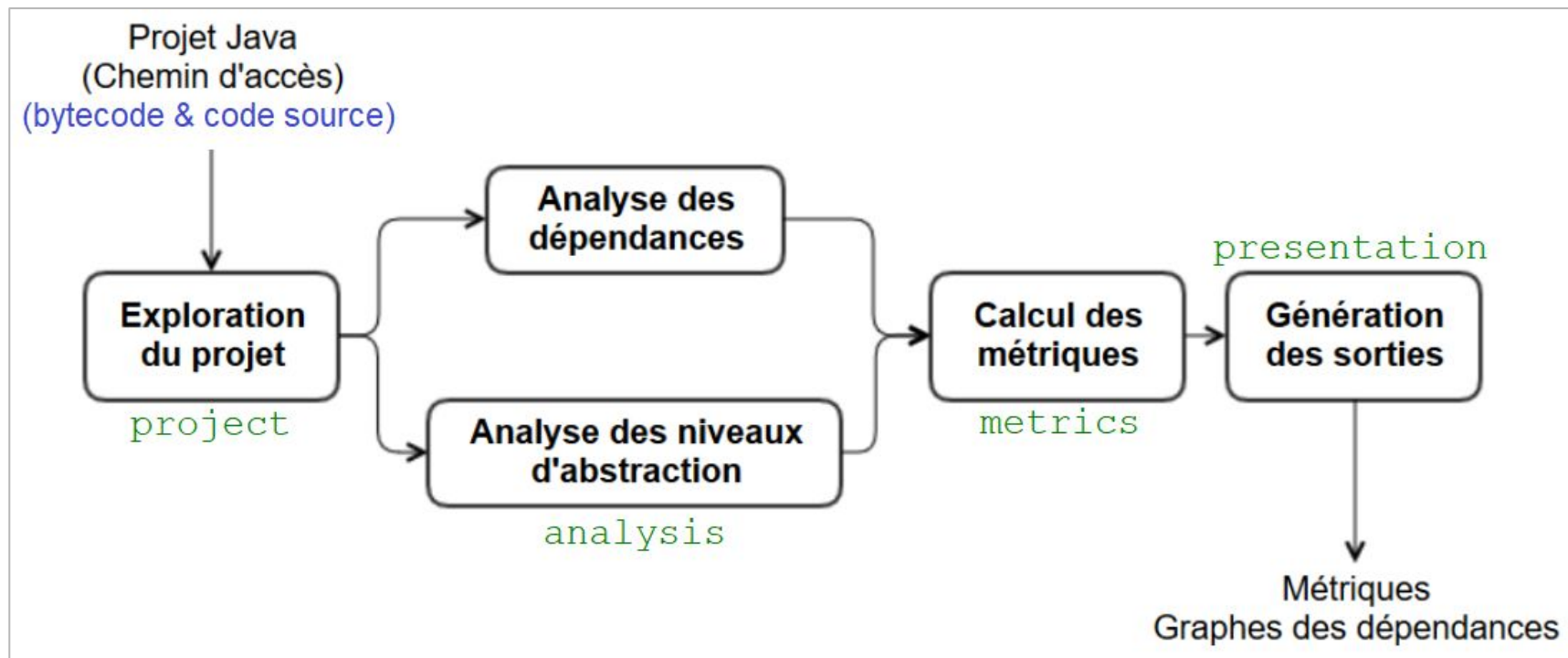
- Distance (**Dn**)



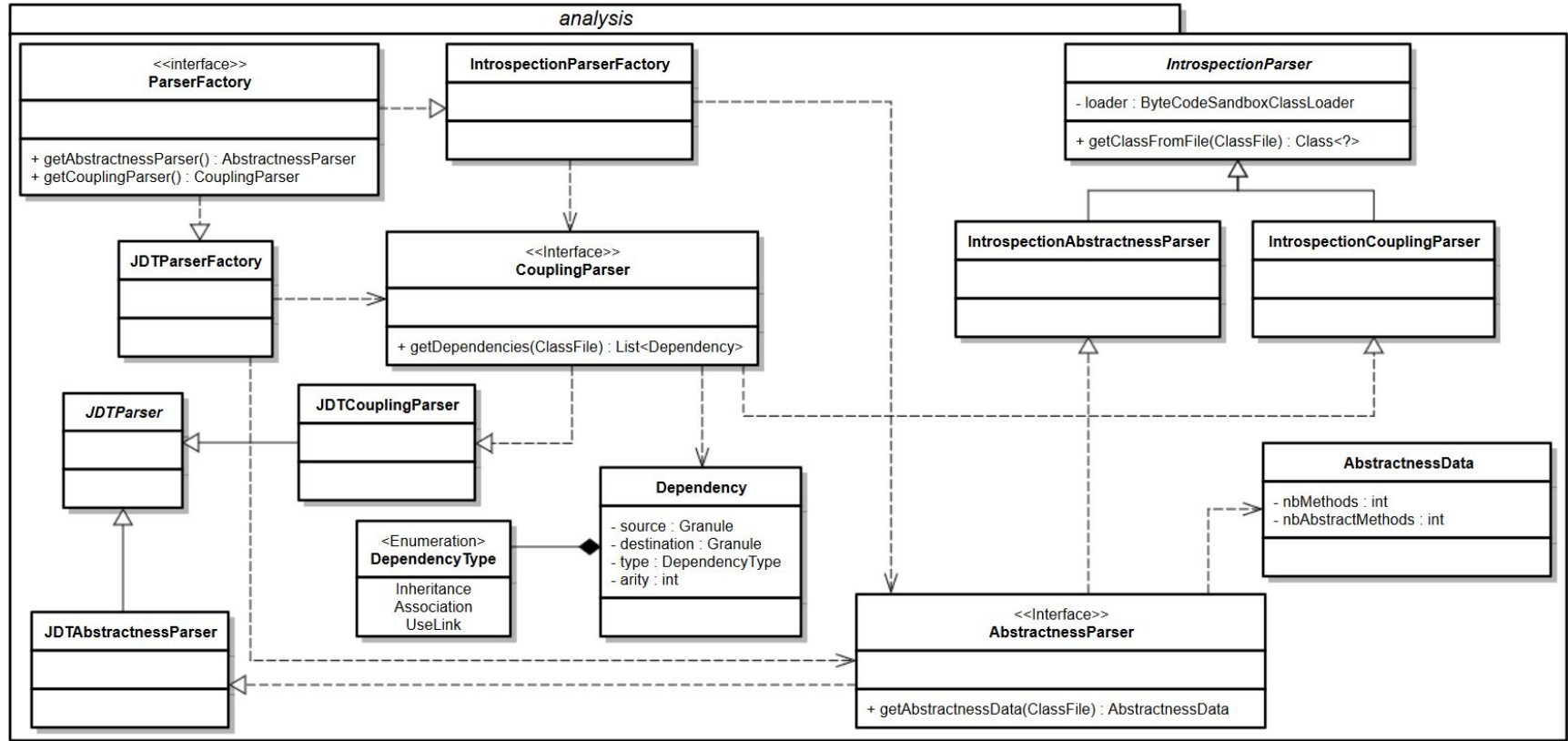
Stable-Abstraction Principle (SAP)

« A component should be as abstract as it is stable. »

Architecture - (1) Pipeline



Architecture - (2) Analyse de code



Méthodes d'analyse

Analyse de bytecode

Pas d'accès à la généricité

Partiellement possible avec *java.lang.reflect*

Majorité des liens à la compilation

→ Moins de travail à l'analyse

Peut extraire des informations différentes
de ce qui est écrit dans le code source

Analyse de code source

Accès à toutes les informations

Nécessite un outil externe (*JDT, JavaParser, ...*)

Nécessite la résolution des liens entre les
éléments du code

Limitations

- Pas de gestion de la multiplicité des dépendances
- Éléments ignorés :
 - Classes internes et anonymes
 - Constructeurs
- Pas de gestion de plusieurs classes par fichier

Comparaison à l'existant

JHawk	JDepend	JMetrics
Propriétaire	Open Source	Open Source
Analyse de code source	Analyse de bytecode	Analyse de code source et de bytecode
Analyse de différentes granularités	Pas d'analyse à l'échelle des classes	Analyse à l'échelle des classes et packages
Combinaison de métriques	Métrique de Martin	Métrique de Martin
Information non disponible	Prise en compte des dépendances externes	Ignore les dépendances externes

Tests - (0) Couverture

Fonctionnalités testées :

- Exploration et création d'une structure représentative du projet
- Analyse de code
- Métriques et calculs
- Graphe, ainsi que primitives et services associés

Tests - (1) Unitaires et Intégration

Vérité terrain annotée

```
@ClassInfo(  
    numberOfMethod = 1,  
    numberOfAbstractMethod = 0,  
    Ca = 2,  
    Ce = 3,  
    I = 0.6,  
    A = 0,  
    Dn = 0.4  
)  
@Dependency(dependencyTo = A.class, type = DependencyType.Inheritance)  
public class B extends A {  
    @Dependency(dependencyTo = C.class, type = DependencyType.Association)  
    private C c;  
    @Dependency(dependencyTo = D.class, type = DependencyType.UseLink)  
    public void doSomething() {  
        D d = new D();  
    }  
}
```

Tests - (2) Performances

Machine : Intel i7 8700k / 16Go RAM

Projet	Nombre de classes	Nombre de packages	Temps
Oracle JDK 1.8	7651	486	10 min 29 sec
Spring Framework	4022	415	4 min 12 sec
JUnit 5	435	58	0 min 15 sec
JMetrics	41	6	0 min 2 sec

Résultats - (0) Données

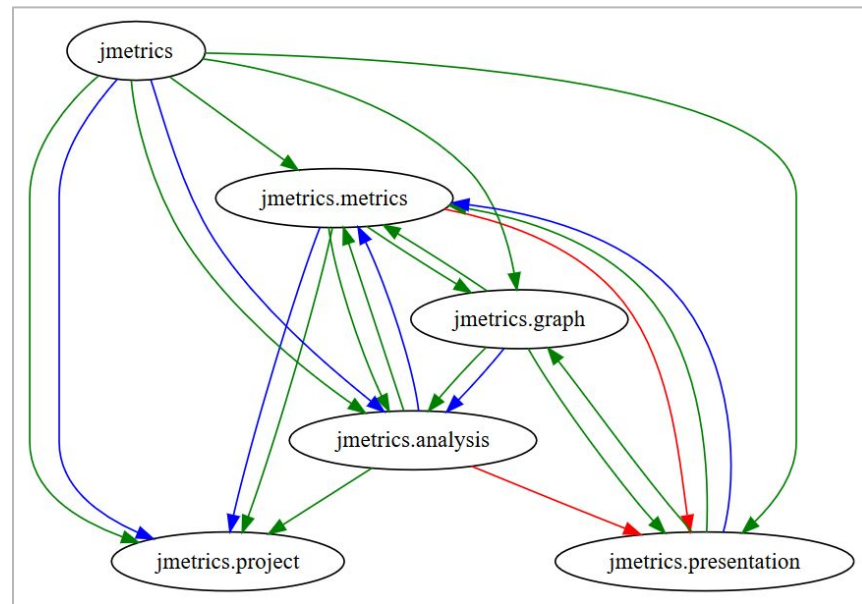
Données extraites (granularités classes et packages) :

- Dépendances
 - Graphe
 - Représentation visuelle (DOT)
 - Matrice d'adjacence (CSV)
 - Liste (CSV)
- Métriques (CSV)

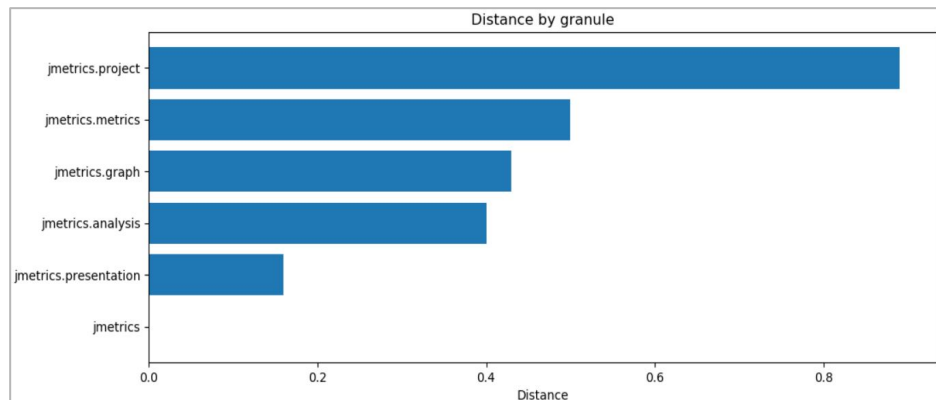
Résultats - (1) Exemple

Métriques et Dépendances

Granule	Ca	Ce	I	A	Dn
jmetrics	0	7	1	0	0
jmetrics.project	5	0	0	0.11	0.89
jmetrics.analysis	6	4	0.4	0.2	0.4
jmetrics.graph	3	4	0.57	0	0.43
jmetrics.metrics	6	6	0.5	0	0.5
jmetrics.presentation	5	4	0.44	0.4	0.16

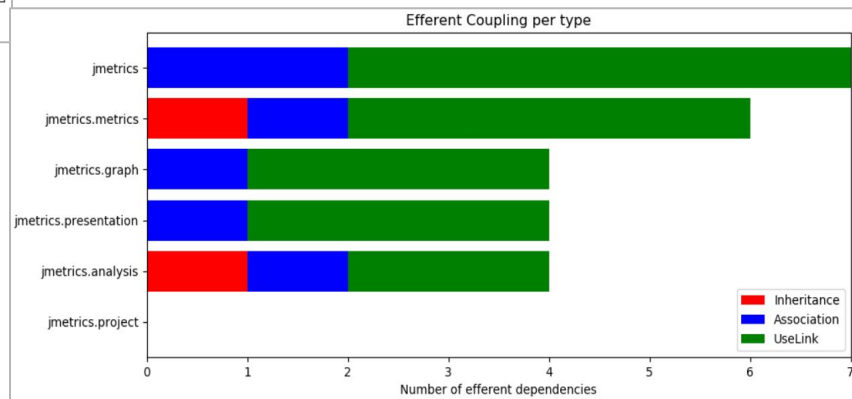


Résultats - (2) Présentation

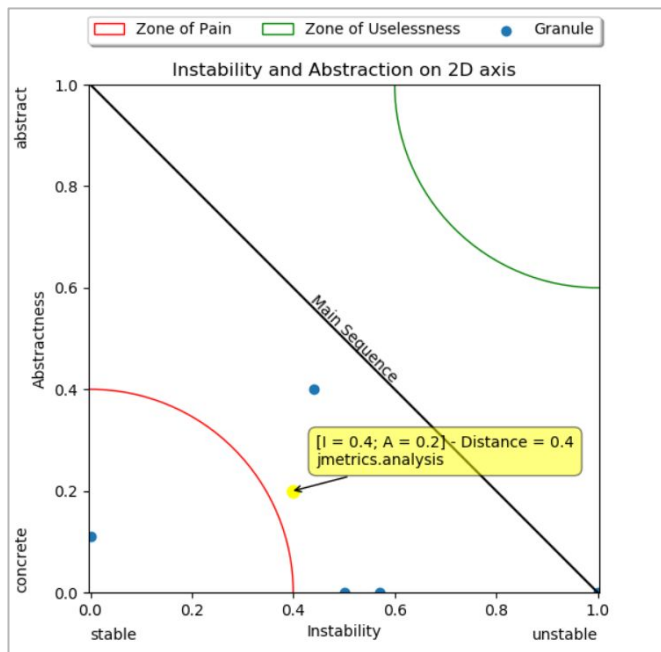


Histogramme du Couplage Efférent
par type de dépendance

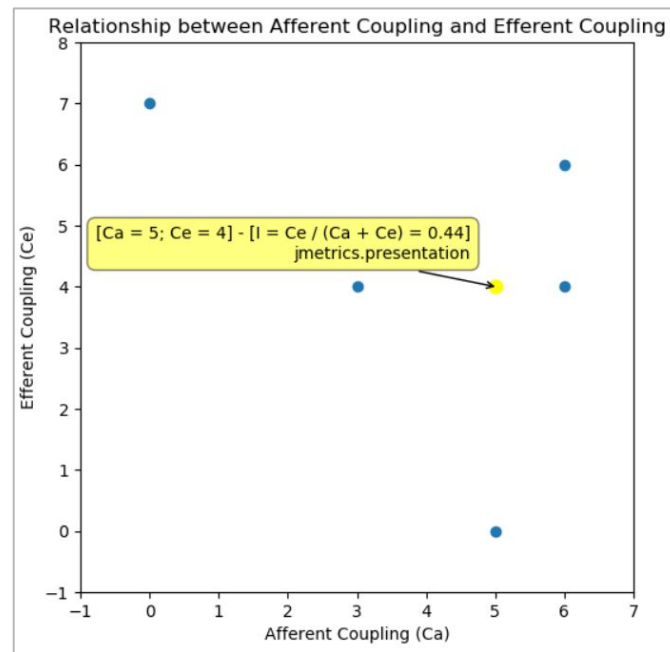
Histogramme de la Distance à la *Main Sequence*



Résultats - (2) Présentation

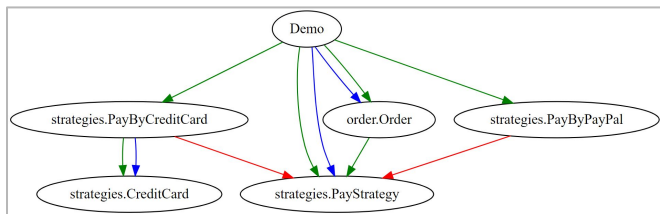


Relation Instabilité / Abstraction sur un axe orthogonal

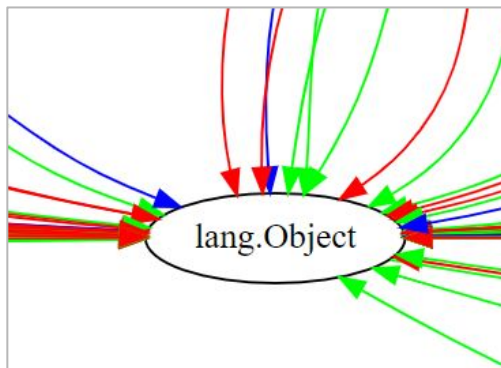


Relation Couplage afférent / efférent sur un axe orthogonal

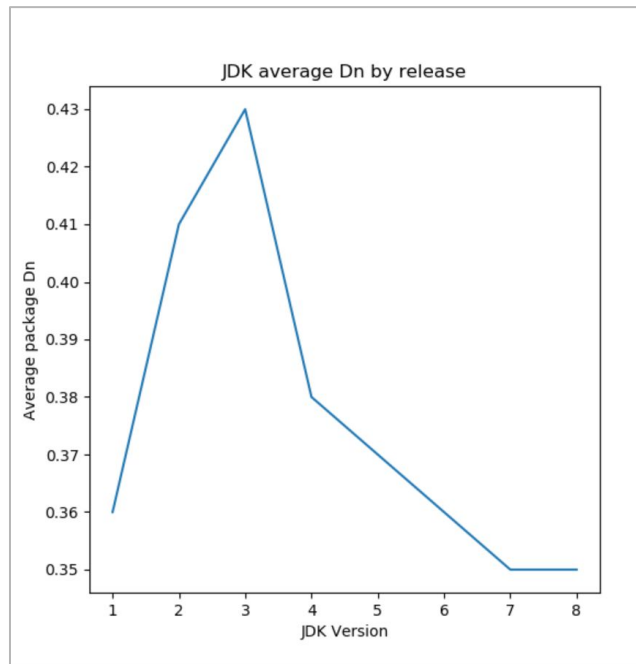
Résultats - (3) Expérimentations



Analyse des Design Patterns
(Graphe de dépendances Strategy)



Noeud granule Object du
graphe de dépendance
java.lang



Evolution **Dn** moyenne releases JDK

Bilan

Un projet se termine...

- Travail effectué
- Difficultés rencontrées
- Résultats encourageants

Perspective

Un projet débute...

- Des améliorations techniques
- Des expérimentations
- Des résultats théoriques
