

# Projet IFT3051

## **Détection semi-automatique des patrons de mauvaise conception dans les architectures orientées objet**

- ❖ PROPOSE PAR  
*Yann-Gaël Guéhéneuc*
- ❖ ENCADRE PAR  
*Naouel Moha*
- ❖ ELABORE PAR  
*Mohammed Amine El Haimer*  
*Najib Tajeddine*



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

# Sommaire

I. Plan.....	3
<u>1. Contexte</u> .....	3
<u>2. Objectifs</u> .....	4
<u>3. Cheminement global du projet</u> .....	5
II. Spécifications des anti-patrons.....	7
<u>1. Introduction</u> .....	7
<b>A. Définition</b> .....	7
<b>B. Les types d'anti-patrons</b> .....	7
III. Définition de la méthodologie.....	13
<u>1. Première itération</u> .....	13
<u>2. Deuxième itération</u> .....	16
<u>3. Troisième itération</u> .....	20
IV. Présentation du livrable final.....	21
<u>1. Détection du blob</u> .....	21
<u>2. Détection du Spaghetti code</u> .....	23
<u>3. Détection du Poltergeists</u> .....	25
<u>4. Détection du Functional Decomposition</u> .....	26
V. Application de la méthodologie.....	27
<u>1. Présentation des résultats</u> .....	27
<u>2. Interprétation des résultats</u> .....	27
VI. L'impact des anti-patrons sur la qualité logiciel.....	30
<u>1. Définitions</u> .....	30
<u>2. la conformité des anti-patrons de conception</u> .....	31
<u>3. Solution proposer en utilisant une action qualité</u> .....	32
Conclusion.....	33
Bibliographie, Références .....	34
Annexe.....	35

---



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

## I. Plan

### 1. Contexte

Dans le cadre de la programmation orientée objet, les règles de la productivité, de l'adaptabilité et de la réutilisabilité doivent obligatoirement être respectées dans la conception et l'élaboration de l'application.

**Est-ce qu'un code fonctionnel est efficace ?** Effectivement un logiciel qui répond au besoin de l'utilisateur n'est pas nécessairement un logiciel dont le code respecte les spécifications de la qualité logiciel.

Notre travail consiste à étudier, à détecter les patrons de mauvaise conception et déterminer leur impact sur la qualité des logiciels, il s'agit de la première étude de ce type qui traite les solutions négatives générées non par des facteurs technologiques mais par des processus humains requérant différentes visions et coopérations pour réaliser une application.

*“Many of the most significant software problems are generated from the human processes that require shared vision, cooperation, and collaboration to realize systems.”*

Thomas J. Mowbray



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

## 2. Objectifs

Le but de cette étude consiste à :

- Définir une forme générale pour chacun des mauvais patrons de conception à travers leurs descriptions, leurs symptômes et leurs méthodes de détection.
- Définir une méthodologie commune pour la détection des mauvais patrons de conception en faisant appel aux métriques.
- Déterminer l'impact des mauvais patrons de conception sur la qualité logiciel.



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

3. Cheminement global du projet

✚ **Étape I :**

- ✓ Lecture du livre de référence :

*AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis* nous a servi comme livre de référence au cours de notre étude et nous a permis de prendre connaissance des différents aspects des mauvais patrons de conception.

- ✓ Élaboration des diagrammes de détection pour chacun des mauvais patrons de conception :

Le recours à la méthode spirale nous a permis de présenter nos résultats, chaque itération représente un raffinement des résultats précédents.

En utilisant des notes collectées dans le livre de référence et des indications de notre encadrant on a pu élaborer des diagrammes par niveau des différentes opérations de détections des mauvais patrons de conception.

- ✓ Fusion de l'ensemble des diagrammes en un seul diagramme pour déterminer une méthodologie commune de détection :

C'est à la dernière itération que l'on a procédé à la fusion des différents diagrammes pour donner une vision globale et commune à la méthodologie de détection.



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

**+ Étape II :**

- ✓ Familiarisation avec les outils de détection : POM, PADL, PADL CLASS CREATOR, CPL.
- ✓ Implantation des détecteurs (classes : BlobDetector, SpaghettiCodeDetector , PolteirgestDetector , FunctionalDecompositionDetector ) des mauvais patrons de conception.
- ✓ En codant en JAVA et en utilisant les métriques logiciels on a pu implanter des classes de détections des mauvais patrons de conception.
- ✓ Application des outils de détection.
- ✓ Interprétation des résultats obtenus.
- ✓ Validation des résultats.

**+ Étape III :**

- ✓ Analyse de l'impact des mauvais patrons de conception sur la qualité logiciel.



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

✚ Étape I :

## II. Spécifications des anti-patrons

### 1. Introduction

#### **A. Définition**

Les anti-patrons identifient les mauvaises conceptions conduisant à une mauvaise qualité du logiciel, ils décrivent une solution pour un problème générant des conséquences négatives dans le cycle de vie du projet.

Ils présentent une forme générale, qui peut être reconnue par des symptômes et exposent un remède détaillé pour résoudre les problèmes du livrable ou du cycle de vie du livrable.

#### **B. Les types d'anti-patrons :**

Les anti-patrons de conception peuvent être engendrés par des programmeurs, architectes et gestionnaires qui n'ont pas assez d'informations ou d'expériences pour résoudre parfaitement un problème particulier ou en appliquant un patron de conception dans un contexte inadéquat.

- Nom : **Les anti-patrons de développement**  
Acteur : Développeur  
Description : Ils présentent les problèmes techniques et les solutions qui concernent les programmeurs.
  
- Nom : **Les anti-patrons d'architecture**  
Acteur : Architecte logiciel  
Description : Ils identifient et résolvent les problèmes communs touchant à l'architecture et à la structure du système.
  
- Nom : **Les anti-patrons de gestion**  
Acteur : Gestionnaire



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

Description : Ils décrivent des problèmes communs dans le processus du livrable et l'organisation du développement.

- ✦ Dans le livre *AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis*, l'auteur a fait référence à plusieurs anti- patrons, par contre notre étude s'est focalisée sur un seul type d'anti-patron qui est celui de développement dont on trouve :

# Blob

## **Définition :**

- Identifié par une classe qui monopolise tout le traitement et qui encapsule toutes les données.
- Programmation structurée

## **Symptômes :**

- Une classe avec un grand nombre d'attributs et d'opérations.
- Cette large classe communique avec des classes de données.
- Cohésion faible dans la large classe.
- L'absence de conception orientée objet (plutôt orientée procédurale).

## **Conséquences :**

- Limitation de la possibilité de modification des systèmes sans affecter d'autres composants.
- Difficulté à être réutilisée et à être testée
- Utilisation énorme de ressources.



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

# Spaghetti code

## **Définition :**

- C'est un système où on trouve peu de structure, le système inclut un petit nombre d'objets avec des méthodes trop grandes qui sont appelées une seule fois.
- Il y a un faible degré d'interaction entre les objets.

## **Symptômes :**

- Des objets nommés comme des processus, les méthodes sont orientées processus.
- Le flux d'exécution est dédié par l'implémentation des objets, non par les clients des objets.
- Relation minimale entre les objets
- Beaucoup de méthodes n'ont pas de paramètres, et utilisent des variables globales ou de classes.

## **Conséquences :**

- Difficulté de réutilisation du code.
- La perte des avantages de la programmation orientée objet.



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

# Functional Decomposition

## **Définition :**

- Quand les programmeurs sont à l'aise avec les « main » routines qui appellent d'autres routines du système, alors ils essayent de transformer les sous routines en classes ignorant ainsi le design orienté objet.

## **Symptômes :**

- Une classe nommée comme des routines (Afficher\_table)
- Tous les attributs sont **Private** et sont utilisés seulement dans la classe.
- Une classe avec une seule action.

## **Conséquences :**

- Violation des principes de la programmation orientée objet rendant difficile la maintenance du logiciel.
- Difficulté de clairement documenter le code.
- Difficulté de réutiliser le code et de le tester.



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

# Poltergeists

## **Définition :**

- Des classes qui ont des responsabilités et des rôles limités dans le système.
- Une classe qui apparaît pendant une courte durée.

## **Symptômes :**

- Redondance dans la navigation du flux.
- Classe sans états.
- Une classe qui apparaît pendant une courte durée.
- Une classe avec une seule action.
- Une classe nommée comme (init\_processus\_1).

## **Conséquences :**

- Utilisation énorme de ressources.

**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

# LavaFlow

## **Définition :**

Le lavaFlow est un anti-patron qui présente des bouts de code ignorés ou des bouts de code dit critiques qu'il ne faut pas toucher.

## **Symptômes :**

- Présence fréquente de variables et de fragments de code injustifiés.
- Manque de documentation des composants critiques du système.
- Présence de code inutilisé.
- Présence des mots « in flux », « to be replaced » et « to do ».
- Des interfaces inexplicables ou obsolètes.

## **Conséquences :**

- Si le lavaFlow n'est pas éliminé il peut se propager en cas de réutilisation du code.
- En cas de propagation du lavaFlow il devient très difficile de documenter le code, et de comprendre son architecture.



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

### III. Définition de la méthodologie

On a utilisé l'approche spirale pour exposer les différents résultats obtenus. Chaque itération représente un raffinement des résultats de l'itération précédente.

#### 1. Première itération

Cette première itération consiste à présenter un découpage des conditions de détection par niveau de chaque anti-patron de conception.

- **BLOB**

**Niveau conception :**

Vérifier la conception.

→ Absence de la conception orientée objet.

Vérifier la communication avec des classes de données.

→ Si la classe communique avec des classes de données.

**Niveau classe :**

Vérifier le nombre d'attributs et de méthodes.

→ Si le nombre d'attributs et de méthodes  $\geq 60$ .

Vérifier la cohésion.

→ Si cohésion faible.



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

- **Spaghetti code**

**Niveau conception :**

Vérifier la relation entre les objets.

→ S'il y a une relation faible.

**Niveau flux d'exécution :**

Vérifier le flux d'exécution.

→ Si le flux est dirigé par l'implémentation des objets non par les clients des objets.

**Niveau classe :**

Vérifier les nomenclatures des objets.

→ Si les objets sont nommés comme des processus.

**Niveau méthode :**

Vérifier les méthodes.

→ Si les méthodes sont orientées processus.

→ Si existence de méthodes sans paramètres.

→ Si les méthodes utilisent des variables globales ou de classes.

- **Functional Decomposition**

**Niveau classe :**

Vérifier la visibilité et l'accessibilité des attributs

→ L'existence de classes avec des attributs **private** utilisés dans la classe.

Vérifier le comportement de la classe.

→ L'existence de classes avec une seule action.

**Niveau méthode :**

Vérifier les nomenclatures des méthodes.

→ Si existence de méthodes nommées comme (Affiche\_table).



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

- **Poltergeists**

**Niveau flux d'exécution :**

Vérifier le flux de navigation.

→ Si le flux de navigation est redondant.

**Niveau classe :**

Vérifier l'état de la classe.

→ L'existence d'une classe sans état.

Vérifier la durée de vie de la classe.

→ Si la durée est courte.

Vérifier le comportement de la classe.

→ L'existence de classes avec une seule action.

- **LavaFlow**

- Trouver fréquemment des variables et des fragments de code injustifiés.
- Manque de documentation des composantes critiques du système.
- Dead code
- La présence des mots « inFlux » et « to be replaced » et « todo »
- Présence des entités et des interfaces obsolètes.

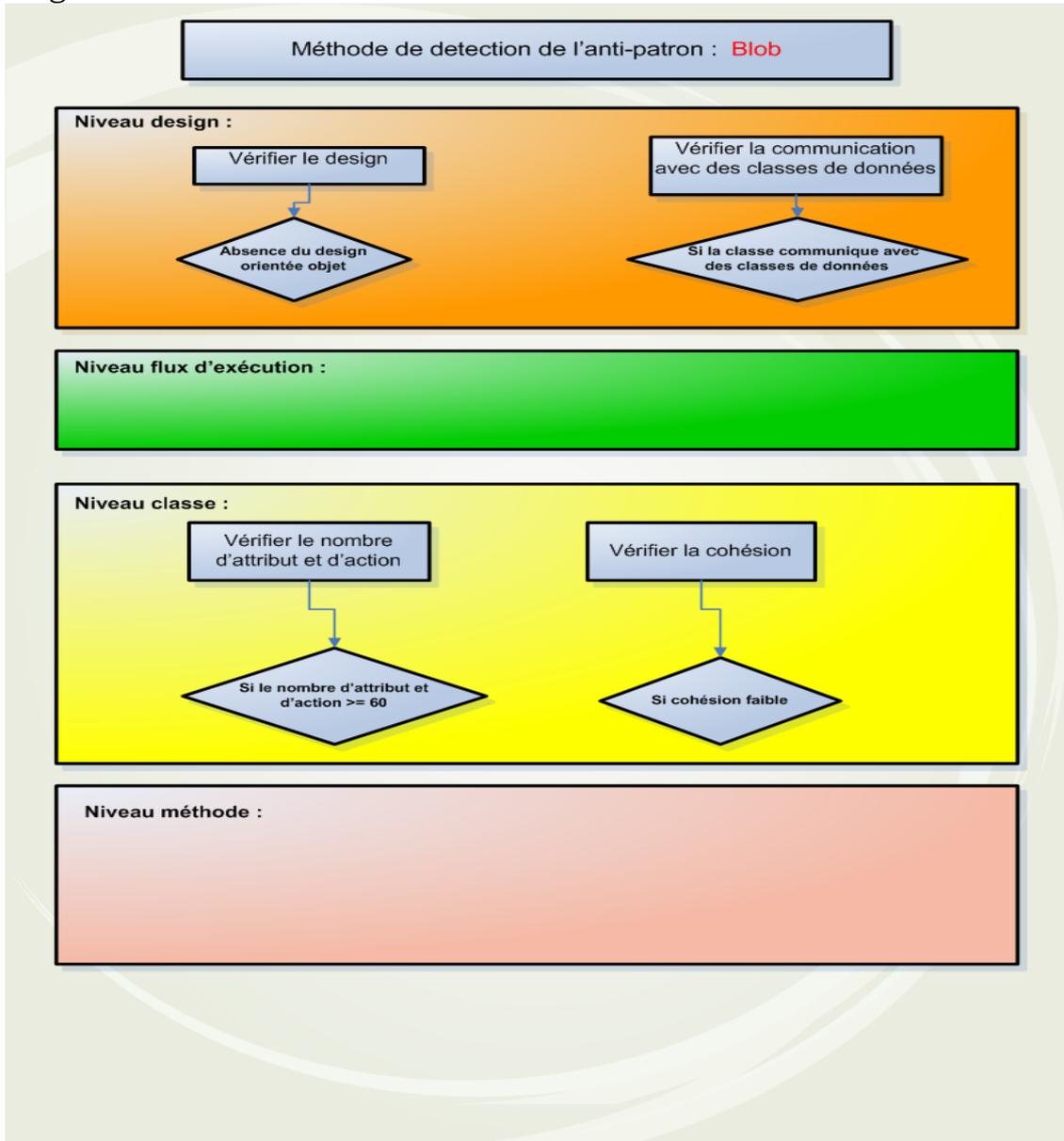


**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

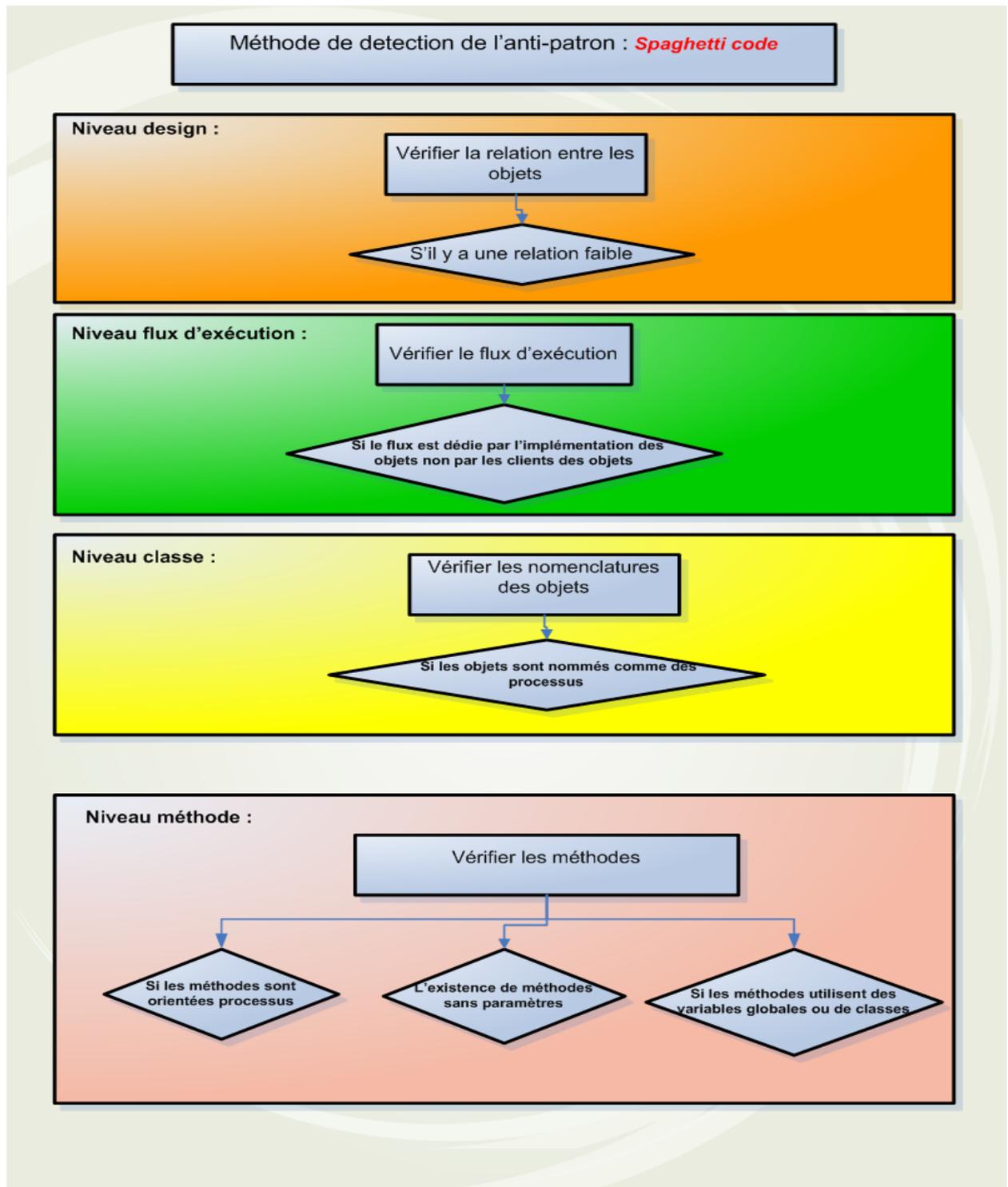
---

## 2. Deuxième itération

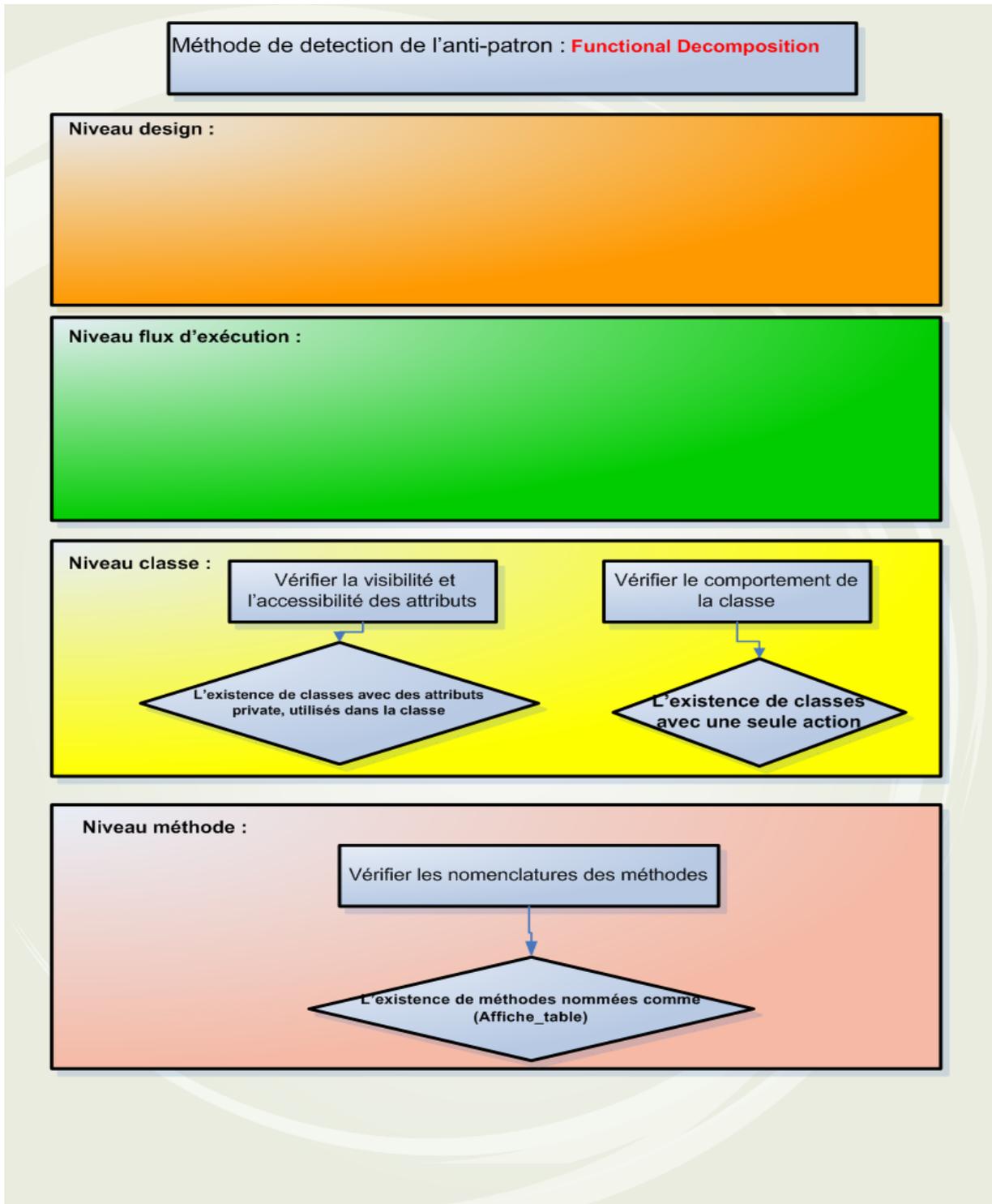
Pour schématiser les différentes conditions de détection on a réalisé les diagrammes suivants :



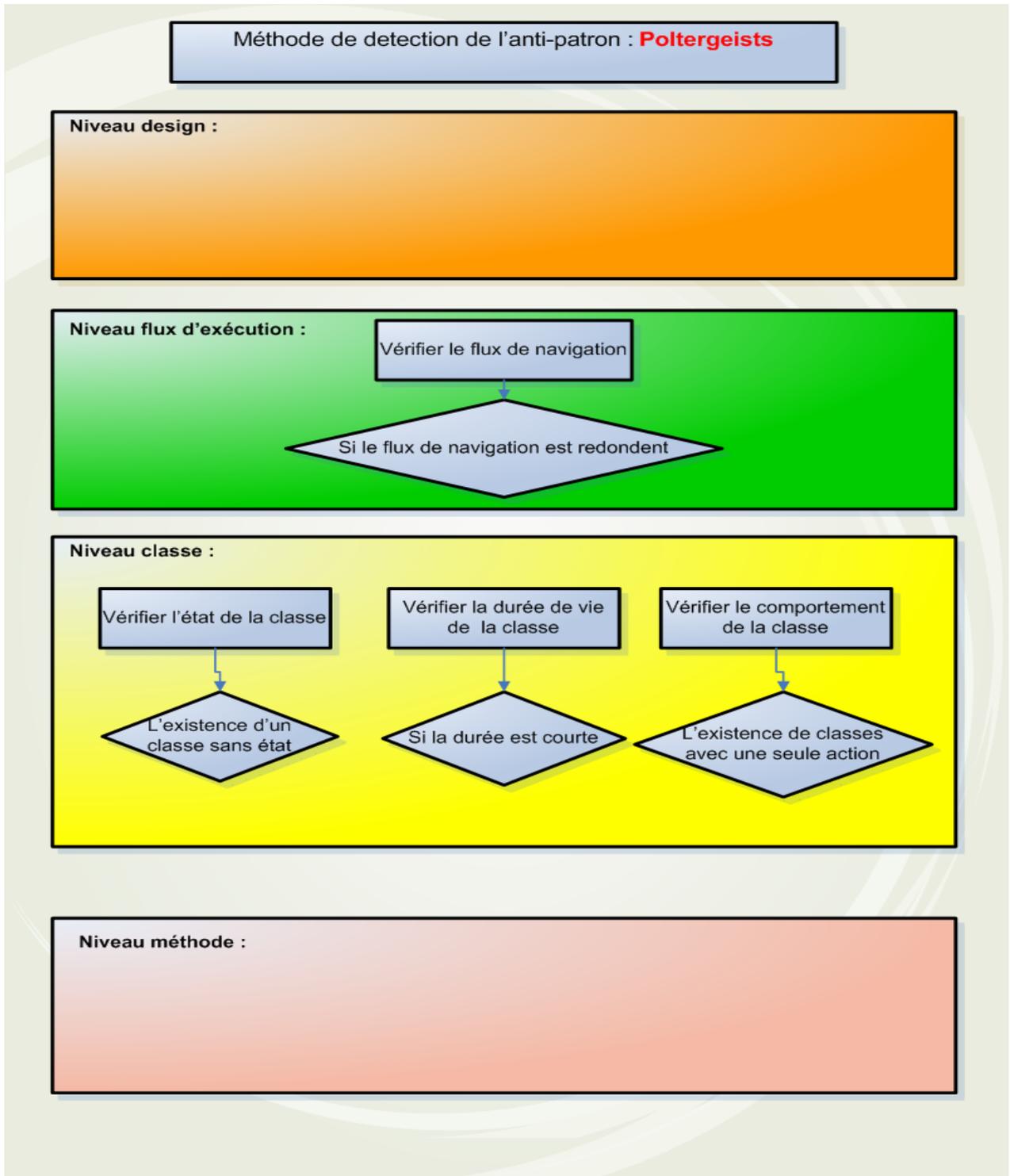
**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**



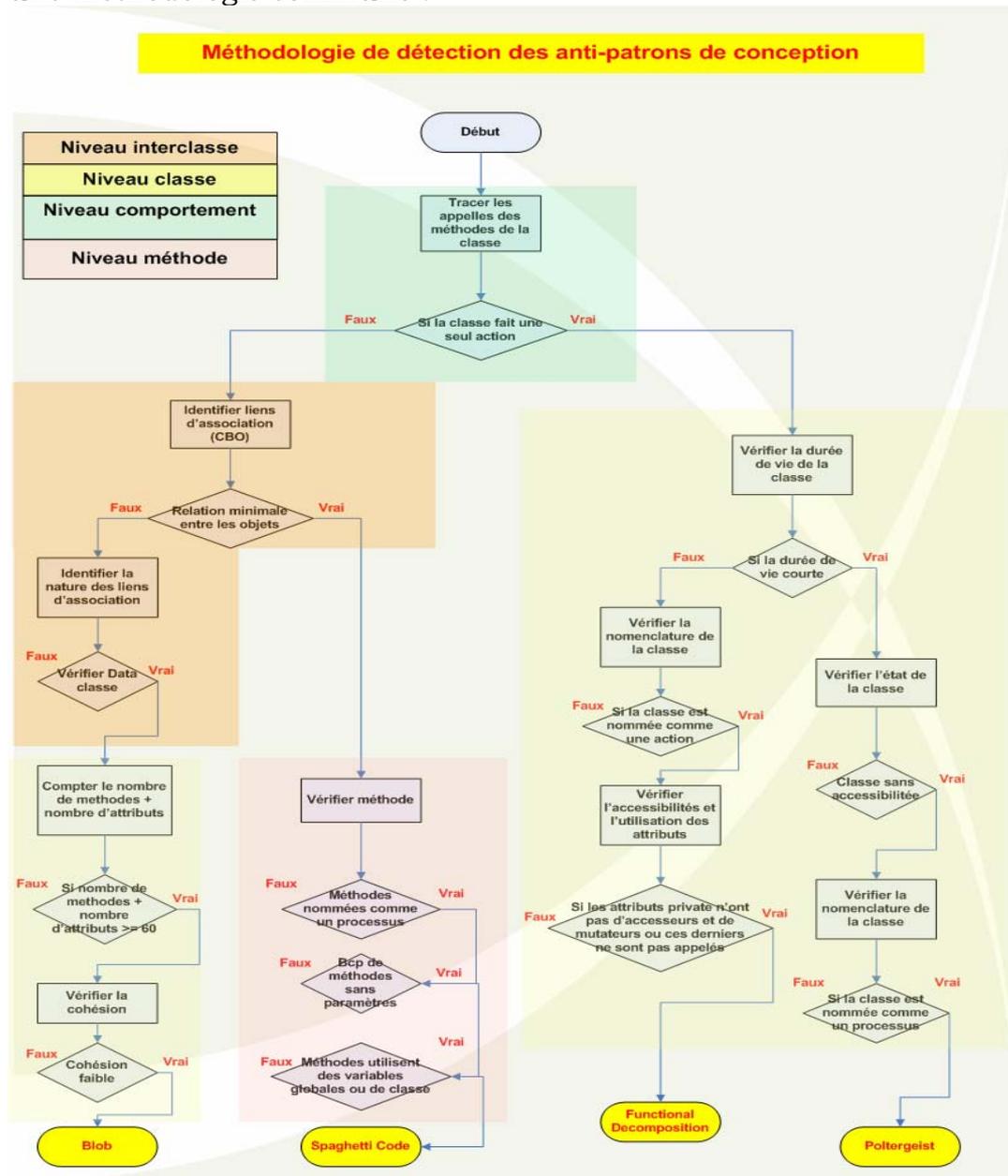
**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**



**Détection semi-automatique des patrons de mauvaise conception dans les architectures orientées-objet**

3. Troisième itération

A Cette itération on a regroupé les différents diagrammes pour présenter une méthodologie commune :



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

✚ Étape II :

#### IV. Présentation du livrable final :

A cette itération on a intégré les métriques logicielles pour faciliter l'implantation des détecteurs (des classes de détection) :

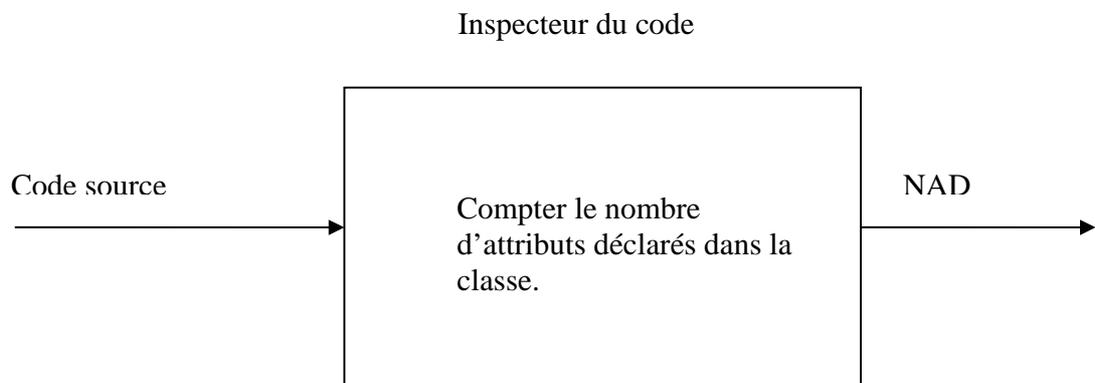
✚ Explication des méthodologies de détections :

##### 1. Détection du blob :

##### ➤ **Étape 1 : Détermination de la large classe**

- Nous avons utilisé deux métriques pour détecter les larges classes :

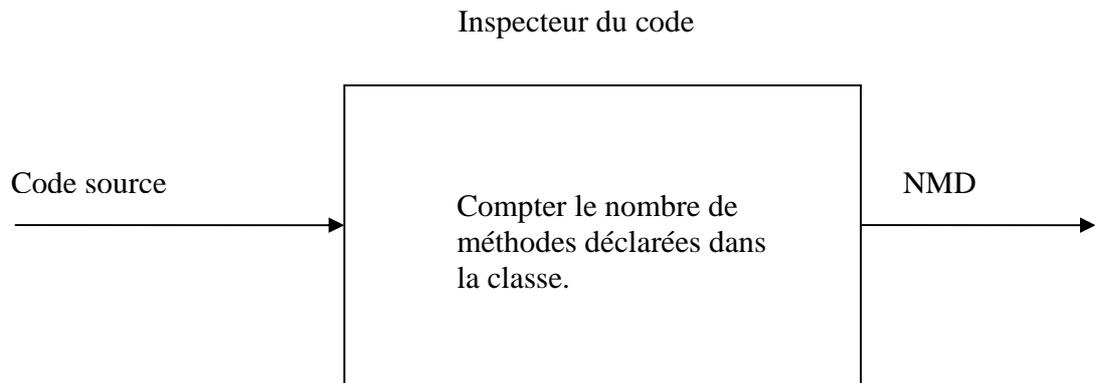
- ✓ NAD (le nombre d'attributs déclarés dans la classe)



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

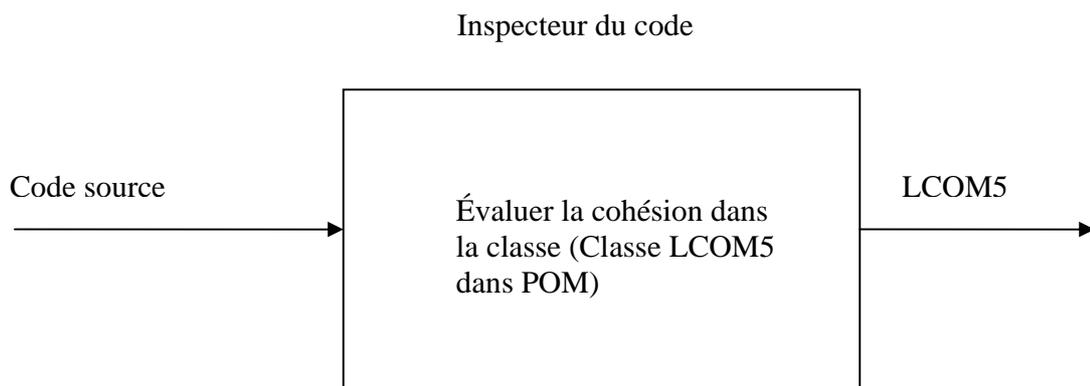
- ✓ NMD (le nombre de méthodes déclarées dans la classe)



- On définit la large classe comme une classe qui a le  $(NAD + NMD) \geq 60$ .

➤ **Étape 2 : Détermination des classes de cohésion faible**

- Nous avons utilisé la métrique LCOM5 :
  - ✓ LCOM5 (Lack of Cohesion in methods)



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

L'ensemble des classes à analyser est obtenu lorsque les deux conditions suivantes sont satisfaites:

**(NAD + NMD) >= 60 && Faible (valeur LCOM5)**

➤ **Étape 3 : Identification des classes de données**

Dans cette étape on utilise le code du CBO modifié dans POM pour obtenir les classes liées avec la large classe.

On définit la valeur NbCDCLC comme le nombre de classes de données qui communiquent avec la large classe.

→Ainsi, on définit le blob comme une classe satisfaisant l'ensemble des conditions dans les trois étapes précédentes.

**(NAD + NMD) >= 60 && Faible (valeur LCOM5) && (NbCDCLC >=1)**

2. Détection du Spaghetti code :

➤ **Étape 1 : Détermination des classes de faible CBO**

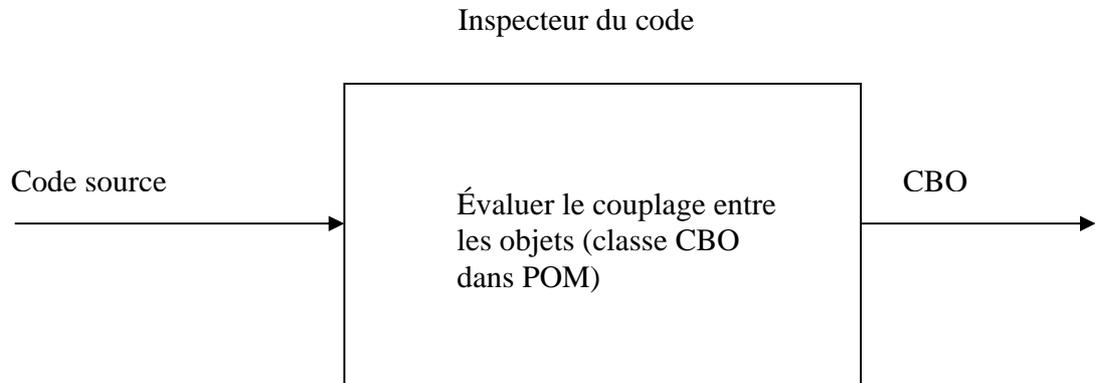
- Nous avons utilisé deux métriques pour détecter les larges classes :

- ✓ CBO (Couplage between objects)



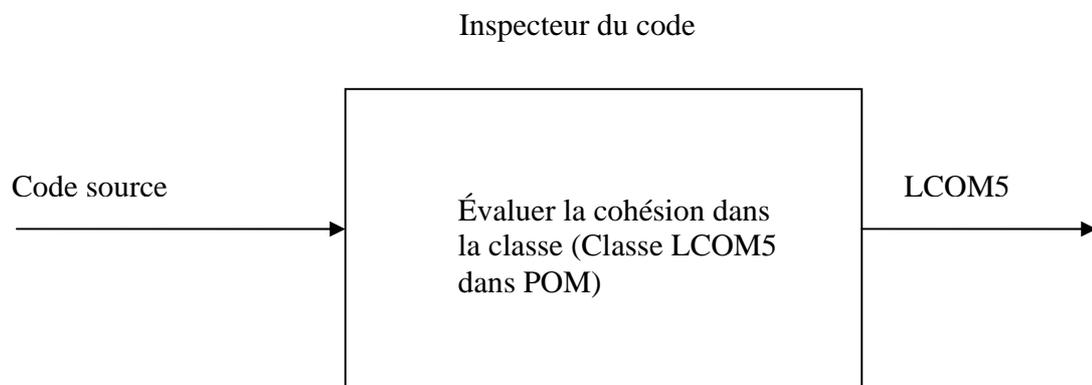
**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---



▪ **Étape 2 : Détermination des classes de cohésion forte**

- Nous avons utilisé la métrique LCOM5 :
  - ✓ LCOM5 (Lack of Cohesion in methods)



- L'ensemble des classes à analyser est obtenu lorsque les deux conditions suivantes sont satisfaites:

**Faible (valeur CBO) && Forte (valeur LCOM5)**

➤ **Étape 3 : Vérification des méthodes**

---



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

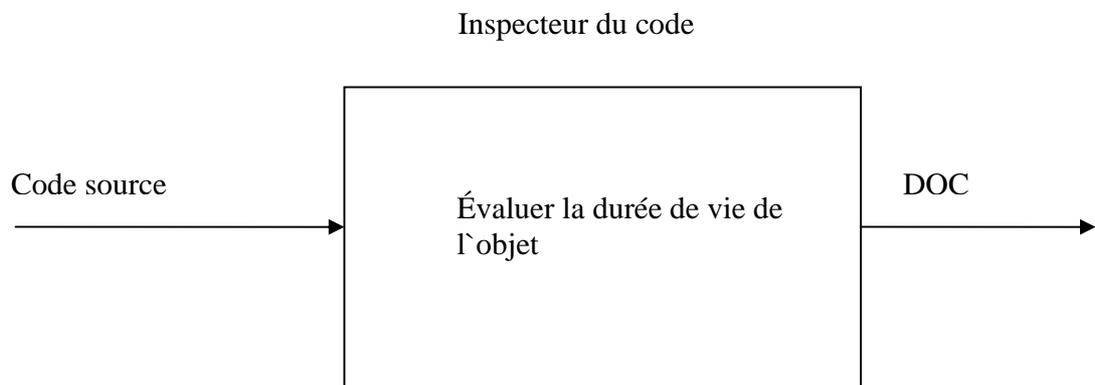
Après l'obtention du résultat de l'étape précédente on récupère toutes les classes qui ont des méthodes sans paramètre et après une vérification visuelle **on détermine les Spaghetti codes**.

### 3. Détection du Poltergeists :

#### ➤ **Étape 1 : Détermination des SDO (Short duration objects)**

- On définit la métrique DOC (duration object calculator), comme la métrique qui détermine la durée de vie d'un objet.

- ✓ DOC (Duration object calculator)



#### ➤ **Étape2 : Déterminer si la classe fait une seule action**

Pour cela on doit vérifier si la classe a un seul point d'entrée ce qui signifie que toutes les méthodes de la classe sont private et une seule méthode est publique.

**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

- L'ensemble des classes à analyser est obtenu par la satisfaction des conditions :

**Minimal(DOC) && classe fait une seule action && Classe sans accessibilité (non public, non private ....)**

On détermine les **Poltergeists** en vérifiant visuellement le nom de la classe (nomme comme un processus).

4. Détection du Functional Decomposition :

➤ **Étape 1 : Déterminer si l'utilisation des attributs est locale**

Pour cela on vérifie si les attributs n'ont pas des accesseurs ou des mutateurs.

➤ **Étape2 : Déterminer si la classe fait une seule action**

Pour cela on doit vérifier si la classe a un seul point d'entrée ce qui signifie que toutes les méthodes de la classe sont private et une seule méthode est publique.

➤ **Étape 3 : Affirmer le Functional Decomposition**

Dans cette étape on vérifie visuellement si la classe a un nom d'une action et on obtient la structure Functional Decomposition.



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

## V. Application de la méthodologie

### 1. Présentation des résultats

**(Voir Annexe)**

### 2. Interprétation des résultats :

#### **Blob**

#### **Remarque :**

On constate que la distribution de l'anti-patron Blob diffère d'un package à l'autre. Ainsi dans le package des interfaces usager, des classes utilitaires et des classes de test, le nombre de Blob est le plus important mais on rappelle que ce dernier est une classe de traitement donc on ne va pas prendre en considération celles-ci.

On constate que notre méthode a des limites car elle varie selon la nature des classes, on peut expliquer ceci par la non disponibilité d'un code intelligent qui peut prédire cela. Effectivement en analysant ArgoUml on a constaté qu'il y avait des classes vérifiant les conditions du Blob alors qu'après une vérification visuelle, il s'est avéré que ces classes ont des structures normales.

D'après les résultats obtenus lors de l'analyse des deux versions du logiciel DrJava ,on a constaté que la structure Blob évolue d'une part et change de structure d'autre part (moins critique).

**Classes utilitaires :** on définit une classe utilitaire comme étant une classe non homogène qui contient des méthodes statiques et qui exécute des services spécifiques.

**Classes interface usager :** on définit classe interface usager comme étant une classe qui gère les composantes graphiques ou aide à le faire.



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

**Suggestions :**

Le modèle MVC est la solution pour séparer les classes d'interface graphique des classes de données d'une part et les classes de traitement d'une autre part.

Si le programmeur décide d'utiliser ce modèle alors on aura moins de difficulté à détecter le Blob.

**Spaghetti-Code**

**Remarque :**

La structure Spaghetti-Code est affectée par le problème de la nature de la classe, effectivement en analysant DrJava on constate la présence abondante des classes d'interface graphiques et des classes de test. On rappelle que le Spaghetti-Code est une classe de traitement avec de grosses méthodes donc on ne va pas prendre en compte ce type de classes.

L'analyse des deux versions de DrJava nous a indiquée que certaines améliorations ont été apportées entre les versions.

**Suggestions :**

On suggère l'utilisation d'un modèle qui pourrait prédire la nature de la classe et faciliterait par la suite la reconnaissance des classes de traitement.



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

### **Poltergeists**

Suite aux suggestions du professeur, on a utilisé le package JDI qui permet le débogage parallèle d'une application. L'utilisation de la fonction de traçage des demandes d'allocation/désallocation de mémoire pour détecter le SDO –Short Duration Object.

Le Poltergeists été le plus difficile à détecter à cause de la condition du SDO (SDO –Short Duration Object) qui n'est pas facile à mettre en évidence. En effet le débogage se faisant en temps réel on a eu plus de problèmes à tracer le SDO car on n'est pas certain d'exécuter le cas d'utilisation faisant appel à ce dernier ou d'avoir un anti-patron LavaFlow susceptible de fausser le résultat.

Ainsi, la seule approche utilisée est de voir la création /destruction d'un objet dans une boucle.

Notre méthode n'est pas assez affinée pour traiter un Poltergeists en raison des limitations de connaissances et d'expérience dans le domaine dynamique des applications.

### **Functional Decomposition**

On a obtenu plusieurs résultats dans le cadre de la détection de l'anti-patron Functional Decomposition , mais on n'a pas pu les valider à cause du manque d'expérience et du raffinement de la méthode de détection.



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

✚ Étape III :

## VI. L'impact des anti-patrons sur la qualité logiciel

### 1. Définitions :

✚ **La Qualité :**

- ✓ Mesure dans laquelle un système, un composant ou un processus satisfait aux exigences énoncées.
  - ✓ Mesure dans laquelle un système, un composant ou un processus satisfait les besoins ou les attentes du client ou de l'utilisateur.
- 
- ✚ L'**adaptabilité** mesure l'aptitude du logiciel à faciliter l'adjonction de nouvelles fonctionnalités ou la modification de fonctionnalités existantes.
  - ✚ la **correction** mesure le degré de conformité par rapport aux spécifications.
  - ✚ l'**efficacité** mesure l'aptitude d'un logiciel à minimiser la consommation des ressources qu'il utilise.
  - ✚ la **maintenabilité** mesure l'aptitude d'un logiciel à faciliter la localisation et la correction d'erreurs résiduelles.
  - ✚ la **réutilisation** mesure l'aptitude du logiciel à une réutilisation de tout ou partie de ses composants dans le cadre d'un autre projet.
  - ✚ la **testabilité** mesure l'aptitude d'un logiciel à faciliter la vérification de son comportement par rapport à des critères de test et de recette.

**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

*2. la conformité des anti-patrons de conception*

Le tableau ci-dessous est un récapitulatif des conséquences des anti-patrons et des clauses non respectées dans la qualité logiciel.

Anti-patron	Conséquences	Clauses non respectées
<b>Blob</b>	<ul style="list-style-type: none"> <li>- les composantes logicielles ne sont pas indépendantes.</li> <li>- Les composantes logicielles ne sont pas réutilisables.</li> <li>- Utilisation énorme de ressources mémoire.</li> </ul>	<ul style="list-style-type: none"> <li>▪ L'adaptabilité</li> <li>▪ Réutilisation</li> <li>▪ Efficacité</li> </ul>
<b>Poltergeists</b>	<ul style="list-style-type: none"> <li>- Utilisation énorme des ressources du processeur</li> </ul>	<ul style="list-style-type: none"> <li>▪ Efficacité</li> </ul>
<b>Spaghetti code</b>	<ul style="list-style-type: none"> <li>- Les composantes logicielles ne sont pas réutilisables.</li> <li>- Les composantes logicielles ne bénéficient pas des avantages de la programmation orientée objet.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Réutilisation</li> <li>▪ L'adaptabilité, Réutilisation</li> </ul>
<b>Functional Decomposition</b>	<ul style="list-style-type: none"> <li>- les composantes logicielles sont difficiles à maintenir.</li> <li>- Les composantes logicielles sont difficiles à documenter</li> <li>- Les composantes logicielles sont difficiles à réutiliser.</li> <li>- Les composantes logicielles sont difficiles à tester.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Maintenabilité</li> <li>▪ Correction</li> <li>▪ Réutilisation</li> <li>▪ Testabilité</li> </ul>
<b>LavaFlow</b>	<ul style="list-style-type: none"> <li>- Il se propage en cas de réutilisation.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Réutilisation</li> </ul>



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

D'après le tableau, on constate que l'impact des anti-patrons sur la qualité logiciel est majeur. Effectivement, beaucoup de clauses de qualités ne sont pas respectées, ce qui engendre des coûts de la non qualité logiciel, ces derniers se présentent sous deux formes :

➤ **Le coût des anomalies internes.**

- Produits déclassés.
- Composantes inemployables.
- Produits à refaire d'une version à l'autre.

➤ **Le coût des anomalies externes**

- Modifications ou adaptations prohibitives.
- Entretien des matériels de maintenance corrective et des compétences.
- Échec commercial : Pertes de clients.

**Conclusion :** on pourrait considérer un anti-patron comme un défaut à cause des pertes qui peut engendrer.

3. Solution proposer en utilisant une action qualité :

L'action de qualité « Inspection » permet de minimiser l'impact des anti-patrons. Le but de l'inspection est de détecter des défauts tôt dans le processus de création du logiciel afin de réduire la reprise et les coûts et augmenter la qualité et la productivité.

On pourrait utiliser cette action pour inspecter la conformité du code source du programmeur avec les principes de qualités et de la programmation orientée objet.



## Conclusion

La richesse du domaine des anti-patrons de conception ne peut être limitée à notre étude.

Ainsi, les résultats obtenus constituent un point de départ pour les prochaines études afin d'apporter des améliorations à la méthode de détection semi-automatique des anti-patrons de conception ce qui nécessite plus d'expérience et d'outils.



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

## Bibliographie, Références

- ❖ *Livre de reference : AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis*
- ❖ [http://icmgworld.com/corp/news/Articles/RS/jan\\_0202.asp](http://icmgworld.com/corp/news/Articles/RS/jan_0202.asp)
- ❖ <http://www.expressitpeople.com/20020318/management3.shtml>
- ❖ <http://www.expressitpeople.com/20020204/management4.shtml>
- ❖ [http://icmgworld.com/corp/news/Articles/RS/jan\\_0302.asp](http://icmgworld.com/corp/news/Articles/RS/jan_0302.asp)
- ❖ Notes de cours LOG330, partie qualité logiciel.



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

## Annexe

**BLOB**

**Logiciel analysé : GanttProject  
Version : v1.10.2**

Large classe	NAD	NMD	Les classes de données	Cohésion
net.sourceforge.ganttproject. <b>ResourceLoadGraphicArea</b>	44.0	44.0	* net.sourceforge.ganttproject.resourceload. ProjectResource	0.9859307348018601
net.sourceforge.ganttproject. <b>GanttOptions</b>	36.0	97.0	* net.sourceforge.ganttproject.io. CSVOptions * net.sourceforge.ganttproject.action. ResourceActionSet * net.sourceforge.ganttproject. GanttPrintable	0.97163742661084
net.sourceforge.ganttproject.gui. <b>GanttTaskPropertiesBean</b>	67.0	34.0	* net.sourceforge.ganttproject.shape. JPaintCombo * net.sourceforge.ganttproject.shape. ShapeConstants * net.sourceforge.ganttproject.shape. ColorConstants	0.9673507477564272
net.sourceforge.ganttproject. <b>GanttGraphicArea</b>	49.0	78.0	* net.sourceforge.ganttproject.resourceload. ProjectResource	0.9798603658809474
net.sourceforge.ganttproject. <b>GanttProject</b>	87.0	125.0	* net.sourceforge.ganttproject.gui. .projectwizard.NewProjectWizard * net.sourceforge. ganttproject.resourceload. ProjectResource net.sourceforge.ganttproject.PrjInfos	0.9806858866674001
net.sourceforge.ganttproject.task. <b>TaskImpl</b>	22.0	48.0	* net.sourceforge.ganttproject. action.ResourceActionSet * net.sourceforge.ganttproject. resourceload.ResourceEvent * net.sourceforge.ganttproject. roles.RoleImpl * net.sourceforge.ganttproject. shape.ShapeConstants	0.9707070691717996



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

* net.sourceforge.ganttproject.task.dependency. constraint.ConstraintImpl * net.sourceforge.ganttproject.task.hierarchy. TaskHierarchyManagerImpl
---

**Logiciel analyse : ArgoUml**  
**Version : suggérée par l'encadrant**

<b>Large classe</b>	<b>NAD</b>	<b>NMD</b>	<b>Les classes de données</b>
org.argouml.model.uml. <b>NSUMLModelImplementation</b>	26.0	37.0	org.argouml.model.uml.CopyHelper org.argouml.model.uml.ExtensionMechanismsHelperImpl
org.argouml.model. <b>uml.CoreHelperImpl</b>  (0.9601449275362319)	4.0	140.0	org.argouml.application.configuration.ConfigurationFactory org.argouml.uml.diagram.ui.CompartmentFigText org.argouml.uml.UUIDHelper org.argouml.uml.ui.SourcePathTableModel org.argouml.cognitive.ui.WizDescription org.argouml.model.uml.ExtensionMechanismsHelperImpl org.argouml.uml.diagram.use_case.ui.UseCaseDiagramRenderer org.argouml.uml.diagram.collaboration.ui.CollabDiagramRenderer org.argouml.ui.explorer.TransferableModelElement org.argouml.uml.diagram.static_structure.ui.ClassDiagramRenderer org.argouml.uml.diagram.sequence.ui.SequenceDiagramRenderer org.argouml.application.security.ArgoJarClassLoader org.argouml.uml.diagram.static_structure.layout. ClassdiagramModelElementFactory org.argouml.uml.cognitive.critics.CrMultipleAgg org.argouml.uml.cognitive.critics.CrMultiComposite org.argouml.uml.diagram.ui.SPFigEdgeModelElement org.argouml.uml.diagram.state.ui.StateDiagramRenderer org.argouml.uml.diagram.ui.ActionAddAssociationRole org.argouml.model.uml.CopyHelper org.argouml.uml.diagram.deployment.ui. DeploymentDiagramRenderer org.argouml.application.security.ArgoAwtExceptionHandler org.argouml.uml.cognitive.critics.CrUnnavigableAssoc
org.argouml.language. java.generator.	24.0	92.0	org.argouml.language.java.generator.CompositeCodePiece



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

<b>JavaRecognizer</b>  (0.9825581386685371)			
org.argouml. cognitive. <b>ToDoList</b>  (0.9502840909090909)	16.0	46.0	org.argouml.application.configuration.ConfigurationFactory org.argouml.uml.diagram.ui.CompartmentFigText org.argouml.cognitive.ToDoListEvent org.argouml.uml.UUIDHelper org.argouml.uml.ui.SourcePathTableModel org.argouml.cognitive.ui.WizDescription org.argouml.uml.diagram.use_case.ui.UseCaseDiagramRenderer org.argouml.uml.diagram.collaboration.ui.CollabDiagramRenderer org.argouml.uml.diagram.static_structure.ui.ClassDiagramRenderer org.argouml.ui.explorer.TransferableModelElement org.argouml.uml.diagram.sequence.ui.SequenceDiagramRenderer org.argouml.application.security.ArgoJarClassLoader org.argouml.uml.diagram.static_structure.layout. ClassdiagramModelElementFactory org.argouml.uml.cognitive.critics.CrMultipleAgg org.argouml.uml.cognitive.critics.CrMultiComposite org.argouml.uml.diagram.ui.SPFigEdgeModelElement org.argouml.uml.diagram.state.ui.StateDiagramRenderer org.argouml.uml.diagram.ui.ActionAddAssociationRole org.argouml.uml.diagram.deployment.ui.DeploymentDiagramRenderer org.argouml.application.security.ArgoAwtExceptionHandler org.argouml.uml.cognitive.critics.CrUnnavigableAssoc
org.argouml. model.uml. <b>CoreFactoryImpl</b>  (0.9294871770991728)	3.0	106.0	org.argouml.application.configuration.ConfigurationFactory org.argouml.uml.diagram.ui.CompartmentFigText org.argouml.uml.UUIDHelper org.argouml.uml.ui.SourcePathTableModel org.argouml.cognitive.ui.WizDescription org.argouml.model.uml.ExtensionMechanismsHelperImpl org.argouml.uml.diagram.use_case.ui.UseCaseDiagramRenderer org.argouml.uml.diagram.collaboration.ui.CollabDiagramRenderer org.argouml.uml.diagram.static_structure.ui.ClassDiagramRenderer org.argouml.ui.explorer.TransferableModelElement org.argouml.uml.diagram.sequence.ui.SequenceDiagramRenderer org.argouml.application.security.ArgoJarClassLoader org.argouml.uml.diagram.static_structure.layout. ClassdiagramModelElementFactory org.argouml.uml.cognitive.critics.CrMultipleAgg



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

			org.argouml.persistence.LastLoadInfo org.argouml.uml.cognitive.critics.CrMultiComposite org.argouml.uml.diagram.ui.SPFigEdgeModelElement org.argouml.uml.diagram.state.ui.StateDiagramRenderer org.argouml.uml.diagram.ui.ActionAddAssociationRole org.argouml.model.uml.CopyHelper org.argouml.uml.diagram.deployment.ui.DeploymentDiagramRenderer org.argouml.application.security.ArgoAwtExceptionHandler org.argouml.uml.cognitive.critics.CrUnnavigableAssoc
org.argouml. uml.reveng.java. <b>JavaLexer</b> (0.9850560389969447)	11.0	78.0	org.argouml.language.java.generator.SimpleCodePiece
org.argouml. uml.generator. <b>ParserDisplay</b> (0.9895178194878236)	9.0	55.0	org.argouml.application.configuration.ConfigurationFactory org.argouml.uml.ui.behavior.collaborations. ActionSetClassifierRoleMultiplicity org.argouml.uml.diagram.ui.CompartmentFigText org.argouml.uml.UUIDHelper org.argouml.uml.ui.SourcePathTableModel org.argouml.uml.generator.PropertySpecialString org.argouml.cognitive.ui.WizDescription org.argouml.uml.diagram.use_case.ui.UseCaseDiagramRenderer org.argouml.uml.diagram.collaboration.ui.CollabDiagramRenderer org.argouml.uml.diagram.static_structure.ui.ClassDiagramRenderer org.argouml.ui.explorer.TransferableModelElement org.argouml.uml.diagram.sequence.ui.SequenceDiagramRenderer org.argouml.application.security.ArgoJarClassLoader org.argouml.uml.ui.foundation.core. ActionSetStructuralFeatureMultiplicity org.argouml.uml.diagram.static_structure.layout. ClassdiagramModelElementFactory org.argouml.uml.ui.behavior.activity_graphs. PropPanelObjectFlowState org.argouml.uml.cognitive.critics.CrMultipleAgg org.argouml.uml.ui.foundation.core. ActionSetAssociationEndMultiplicity org.argouml.uml.cognitive.critics.CrMultiComposite org.argouml.uml.diagram.ui.SPFigEdgeModelElement org.argouml.uml.diagram.state.ui.StateDiagramRenderer org.argouml.uml.diagram.ui.ActionAddAssociationRole



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

			org.argouml.uml.diagram.deployment.ui.DeploymentDiagramRenderer org.argouml.application.security.ArgoAwtExceptionHandler org.argouml.uml.cognitive.critics.CrUnnavigableAssoc
org.argouml- ui. <b>ProjectBrowser</b> (0.9770531391517984)	23.0	39.0	org.argouml.application.configuration.ConfigurationFactory org.argouml.uml.diagram.ui.CompartmentFigText org.argouml.uml.UUIDHelper org.argouml.uml.ui.SourcePathTableModel org.argouml.cognitive.ui.WizDescription org.argouml.uml.diagram.use_case.ui.UseCaseDiagramRenderer org.argouml.uml.diagram.collaboration.ui.CollabDiagramRenderer org.argouml.uml.diagram.static_structure.ui.ClassDiagramRenderer org.argouml.ui.explorer.TransferableModelElement org.argouml.uml.diagram.sequence.ui.SequenceDiagramRenderer org.argouml.application.security.ArgoJarClassLoader org.argouml.ui.SplashScreen org.argouml.uml.diagram.static_structure.layout. ClassdiagramModelElementFactory org.argouml.uml.cognitive.critics.CrMultipleAgg org.argouml.uml.cognitive.critics.CrMultiComposite org.argouml.uml.diagram.ui.SPFigEdgeModelElement org.argouml.uml.diagram.state.ui.StateDiagramRenderer org.argouml.uml.diagram.ui.ActionAddAssociationRole org.argouml.uml.diagram.deployment.ui.DeploymentDiagramRenderer org.argouml.application.security.ArgoAwtExceptionHandler org.argouml.uml.cognitive.critics.CrUnnavigableAssoc
org.argouml.uml. cognitive.critics. <b>Init</b> (0.10526307672262192)	95.0	3.0	org.argouml.application.configuration.ConfigurationFactory org.argouml.uml.diagram.ui.CompartmentFigText org.argouml.uml.UUIDHelper org.argouml.uml.ui.SourcePathTableModel org.argouml.cognitive.ui.WizDescription org.argouml.uml.diagram.use_case.ui.UseCaseDiagramRenderer org.argouml.uml.diagram.collaboration.ui.CollabDiagramRenderer org.argouml.ui.explorer.TransferableModelElement org.argouml.uml.diagram.static_structure.ui.ClassDiagramRenderer org.argouml.uml.diagram.sequence.ui.SequenceDiagramRenderer org.argouml.application.security.ArgoJarClassLoader org.argouml.uml.diagram.static_structure.layout. ClassdiagramModelElementFactory org.argouml.uml.cognitive.critics.CrMultipleAgg org.argouml.uml.cognitive.critics.CrMultiComposite



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

			org.argouml.uml.diagram.ui.SPFigEdgeModelElement org.argouml.uml.diagram.state.ui.StateDiagramRenderer org.argouml.uml.diagram.ui.ActionAddAssociationRole org.argouml.uml.diagram.deployment.ui.DeploymentDiagramRenderer org.argouml.application.security.ArgoAwtExceptionHandler org.argouml.uml.cognitive.critics.CrUnnavigableAssoc
org.argouml. model.uml. <b>NSUMLModelFacade</b> (0.9968553457245137)	3.0	320.0	org.argouml.application.configuration.ConfigurationFactory org.argouml.uml.diagram.ui.CompartmentFigText org.argouml.uml.UUIDHelper org.argouml.uml.ui.SourcePathTableModel org.argouml.cognitive.ui.WizDescription org.argouml.model.uml.ExtensionMechanismsHelperImpl org.argouml.uml.diagram.use_case.ui.UseCaseDiagramRenderer org.argouml.uml.diagram.collaboration.ui.CollabDiagramRenderer org.argouml.uml.diagram.static_structure.ui.ClassDiagramRenderer org.argouml.ui.explorer.TransferableModelElement org.argouml.uml.diagram.sequence.ui.SequenceDiagramRenderer org.argouml.application.security.ArgoJarClassLoader org.argouml.uml.ui.behavior.common_behavior.ActionNewReception org.argouml.uml.diagram.static_structure.layout. ClassdiagramModelElementFactory org.argouml.uml.cognitive.critics.CrMultipleAgg org.argouml.persistence.LastLoadInfo org.argouml.uml.cognitive.critics.CrMultiComposite org.argouml.uml.diagram.ui.SPFigEdgeModelElement org.argouml.uml.diagram.state.ui.StateDiagramRenderer org.argouml.uml.diagram.ui.ActionAddAssociationRole org.argouml.model.uml.CopyHelper org.argouml.uml.diagram.deployment.ui.DeploymentDiagramRenderer org.argouml.application.security.ArgoAwtExceptionHandler org.argouml.uml.cognitive.critics.CrUnnavigableAssoc
org.argouml. kernel. <b>Project</b> (0.9588675197595969)	24.0	81.0	org.argouml.application.configuration.ConfigurationFactory org.argouml.persistence.ProgressEvent org.argouml.uml.diagram.ui.CompartmentFigText org.argouml.uml.UUIDHelper org.argouml.uml.ui.SourcePathTableModel org.argouml.cognitive.ui.WizDescription org.argouml.uml.diagram.use_case.ui.UseCaseDiagramRenderer org.argouml.uml.diagram.collaboration.ui.CollabDiagramRenderer org.argouml.ui.explorer.TransferableModelElement



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

			org.argouml.uml.diagram.static_structure.ui.ClassDiagramRenderer org.argouml.uml.diagram.sequence.ui.SequenceDiagramRenderer org.argouml.application.security.ArgoJarClassLoader org.argouml.application.ArgoVersion org.argouml.uml.diagram.static_structure.layout. ClassdiagramModelElementFactory org.argouml.uml.cognitive.critics.CrMultipleAgg org.argouml.ocl.ArgoFacade org.argouml.util.ClassFileFilter org.argouml.uml.cognitive.critics.CrMultiComposite org.argouml.uml.diagram.ui.SPFigEdgeModelElement org.argouml.uml.diagram.state.ui.StateDiagramRenderer org.argouml.uml.diagram.ui.ActionAddAssociationRole org.argouml.uml.diagram.deployment.ui.DeploymentDiagramRenderer org.argouml.application.security.ArgoAwtExceptionHandler org.argouml.uml.cognitive.critics.CrUnnavigableAssoc
org.argouml.uml: diagram.ui. <b>FigEdgeModelElement</b> (0.9812499994101623)	10.0	51.0	<del>org.argouml.application.configuration.ConfigurationFactory</del> <del>org.argouml.uml.diagram.ui.CompartmentFigText</del> <del>org.argouml.uml.UUIDHelper</del> <del>org.argouml.uml.ui.SourcePathTableModel</del> <del>org.argouml.uml.generator.PropertySpecialString</del> <del>org.argouml.cognitive.ui.WizDescription</del> <del>org.argouml.uml.ui.UMLTreeCellRenderer</del> <del>org.argouml.uml.diagram.use_case.ui.UseCaseDiagramRenderer</del> <del>org.argouml.uml.diagram.collaboration.ui.CollabDiagramRenderer</del> <del>org.argouml.uml.diagram.static_structure.ui.ClassDiagramRenderer</del> <del>org.argouml.ui.explorer.TransferableModelElement</del> <del>org.argouml.uml.diagram.sequence.ui.SequenceDiagramRenderer</del> <del>org.argouml.application.security.ArgoJarClassLoader</del> <del>org.argouml.language.java.generator.InterfaceCodePiece</del> <del>org.argouml.uml.diagram.ui.FigNameCompartment</del> <del>org.argouml.uml.diagram.static_structure.layout.          ClassdiagramModelElementFactory</del> <del>org.argouml.uml.cognitive.critics.CrMultipleAgg</del> <del>org.argouml.language.java.generator.OperationCodePiece</del> <del>org.argouml.uml.cognitive.critics.CrMultiComposite</del> <del>org.argouml.uml.diagram.ui.SPFigEdgeModelElement</del> <del>org.argouml.uml.diagram.state.ui.StateDiagramRenderer</del> <del>org.argouml.uml.diagram.ui.ActionAddAssociationRole</del> <del>org.argouml.language.java.generator.ClassCodePiece</del>



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

			org.argouml.uml.diagram.deployment.ui.DeploymentDiagramRenderer org.argouml.application.security.ArgoAwtExceptionHandler org.argouml.uml.cognitive.critics.CrUnnavigableAssoc
org.argouml.language. java.generator. <b>JavaLexer</b> (0.9836601304876453)	9.0	73.0	org.argouml.language.java.generator.SimpleCodePiece
org.argouml. cognitive.critics. <b>Critic</b> (0.9893048120772138)	33.0	70.0	org.argouml.application.configuration.ConfigurationFactory org.argouml.persistence.ProgressEvent org.argouml.uml.diagram.ui.CompartmentFigText org.argouml.uml.UIDHelper org.argouml.uml.ui.SourcePathTableModel org.argouml.cognitive.ui.WizDescription org.argouml.uml.diagram.use_case.ui.UseCaseDiagramRenderer org.argouml.uml.diagram.collaboration.ui.CollabDiagramRenderer org.argouml.ui.explorer.TransferableModelElement org.argouml.uml.diagram.static_structure.ui.ClassDiagramRenderer org.argouml.uml.diagram.sequence.ui.SequenceDiagramRenderer org.argouml.application.security.ArgoJarClassLoader org.argouml.uml.diagram.static_structure.layout. ClassDiagramModelElementFactory org.argouml.uml.cognitive.critics.CrMultipleAgg org.argouml.util.ClassFileFilter org.argouml.uml.cognitive.critics.CrMultiComposite org.argouml.uml.diagram.ui.SPFigEdgeModelElement org.argouml.uml.diagram.state.ui.StateDiagramRenderer org.argouml.uml.diagram.ui.ActionAddAssociationRole org.argouml.uml.diagram.deployment.ui.DeploymentDiagramRenderer org.argouml.application.security.ArgoAwtExceptionHandler org.argouml.uml.cognitive.critics.CrUnnavigableAssoc
org.argouml.uml. diagram.ui. <b>FigNodeModelElement</b> (0.9844155829254683)	30.0	80.0	org.argouml.uml.ui.behavior.collaborations. ActionSetClassifierRoleMultiplicity org.argouml.uml.diagram.ui.CompartmentFigText org.argouml.uml.generator.PropertySpecialString org.argouml.cognitive.ui.WizDescription org.argouml.uml.diagram.collaboration.ui.CollabDiagramRenderer org.argouml.uml.diagram.static_structure.ui.ClassDiagramRenderer org.argouml.uml.diagram.sequence.ui.SequenceDiagramRenderer org.argouml.language.java.generator.InterfaceCodePiece



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

			<p>org.argouml.uml.diagram.static_structure.layout. ClassDiagramModelElementFactory org.argouml.uml.cognitive.critics.CrMultipleAgg org.argouml.uml.ui.foundation.core. ActionSetAssociationEndMultiplicity org.argouml.uml.diagram.state.ui.StateDiagramRenderer org.argouml.language.java.generator.ClassCodePiece org.argouml.uml.diagram.deployment.ui.DeploymentDiagramRenderer org.argouml.application.security.ArgoAwtExceptionHandler org.argouml.application.configuration.ConfigurationFactory org.argouml.uml.diagram.ui.FigFeaturesCompartment org.argouml.uml.UIIDHelper org.argouml.uml.ui.SourcePathTableModel org.argouml.uml.ui.UMLTreeCellRenderer org.argouml.uml.diagram.use_case.ui.UseCaseDiagramRenderer org.argouml.ui.explorer.TransferableModelElement org.argouml.application.security.ArgoJarClassLoader org.argouml.uml.diagram.ui.FigNameCompartment org.argouml.uml.ui.foundation.core. ActionSetStructuralFeatureMultiplicity org.argouml.uml.diagram.ui.FigNote org.argouml.language.java.generator.OperationCodePiece org.argouml.uml.cognitive.critics.CrMultiComposite org.argouml.uml.diagram.ui.SPFigEdgeModelElement org.argouml.uml.diagram.ui.ActionAddAssociationRole org.argouml.uml.cognitive.critics.CrUnnavigableAssoc</p>
<p>org.argouml. language. java.generator.<b>GeneratorJava</b>  (0.9529569873646382)</p>	12.0	64.0	<p>org.argouml.application.configuration.ConfigurationFactory org.argouml.uml.ui.behavior.collaborations. ActionSetClassifierRoleMultiplicity org.argouml.uml.diagram.ui.CompartmentFigText org.argouml.uml.UIIDHelper org.argouml.uml.ui.SourcePathTableModel org.argouml.cognitive.ui.WizDescription org.argouml.uml.diagram.use_case.ui.UseCaseDiagramRenderer org.argouml.uml.diagram.collaboration.ui.CollabDiagramRenderer org.argouml.uml.diagram.static_structure.ui.ClassDiagramRenderer org.argouml.ui.explorer.TransferableModelElement org.argouml.uml.diagram.sequence.ui.SequenceDiagramRenderer org.argouml.application.security.ArgoJarClassLoader org.argouml.language.java.generator.InterfaceCodePiece</p>



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

		<p>org.argouml.application.ArgoVersion  org.argouml.uml.ui.foundation.core.  ActionSetStructuralFeatureMultiplicity  org.argouml.uml.diagram.static_structure.layout.  ClassdiagramModelElementFactory  org.argouml.uml.ui.behavior.activity_graphs.  PropPanelObjectFlowState  org.argouml.uml.cognitive.critics.CrMultipleAgg  org.argouml.uml.ui.foundation.core.  ActionSetAssociationEndMultiplicity  org.argouml.language.java.generator.ClassifierEndCodePiece  org.argouml.language.java.generator.OperationCodePiece  org.argouml.uml.cognitive.critics.CrMultiComposite  org.argouml.uml.diagram.ui.SPFigEdgeModelElement  org.argouml.uml.diagram.state.ui.StateDiagramRenderer  org.argouml.uml.diagram.ui.ActionAddAssociationRole  org.argouml.language.java.generator.ClassCodePiece  org.argouml.uml.diagram.deployment.ui.  DeploymentDiagramRenderer  org.argouml.language.java.generator.PackageCodePiece  org.argouml.application.security.ArgoAwtExceptionHandler  org.argouml.uml.cognitive.critics.CrUnnavigableAssoc</p>
<p>org.argouml.  application.api.<b>Notation</b>  (0.9793340974112055)</p>	<p>13.0 69.0</p>	<p>org.argouml.application.configuration.ConfigurationFactory  org.argouml.uml.ui.behavior.collaborations.  ActionSetClassifierRoleMultiplicity  org.argouml.uml.diagram.ui.CompartmentFigText  org.argouml.uml.UUIDHelper  org.argouml.uml.ui.SourcePathTableModel  org.argouml.cognitive.ui.WizDescription  org.argouml.uml.diagram.use_case.ui.UseCaseDiagramRenderer  org.argouml.uml.diagram.collaboration.ui.CollabDiagramRenderer  org.argouml.uml.diagram.static_structure.ui.ClassDiagramRenderer  org.argouml.ui.explorer.TransferableModelElement  org.argouml.uml.diagram.sequence.ui.SequenceDiagramRenderer  org.argouml.application.security.ArgoJarClassLoader  org.argouml.uml.ui.foundation.core.  ActionSetStructuralFeatureMultiplicity  org.argouml.uml.diagram.static_structure.layout.  ClassdiagramModelElementFactory  org.argouml.uml.ui.behavior.activity_graphs.</p>



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

			PropPanelObjectFlowState org.argouml.uml.cognitive.critics.CrMultipleAgg org.argouml.uml.ui.foundation.core. ActionSetAssociationEndMultiplicity org.argouml.uml.cognitive.critics.CrMultiComposite org.argouml.uml.diagram.ui.SPFigEdgeModelElement org.argouml.uml.diagram.state.ui.StateDiagramRenderer org.argouml.uml.diagram.ui.ActionAddAssociationRole org.argouml.uml.diagram.deployment.ui. DeploymentDiagramRenderer org.argouml.application.security.ArgoAwtExceptionHandler org.argouml.uml.cognitive.critics.CrUnnavigableAssoc
org.argouml.uml. diagram.sequence.ui. <b>FigObject</b> (0.9488372054210928)	15.0	46.0	org.argouml.uml.diagram.sequence.ObjectNode
org.argouml. cognitive. <b>Designer</b> (0.9809964036537429)	33.0	61.0	org.argouml.application.configuration.ConfigurationFactory org.argouml.uml.diagram.ui.CompartmentFigText org.argouml.uml.UUIDHelper org.argouml.uml.ui.SourcePathTableModel org.argouml.cognitive.ui.WizDescription org.argouml.uml.diagram.use_case.ui.UseCaseDiagramRenderer org.argouml.uml.diagram.collaboration.ui.CollabDiagramRenderer org.argouml.uml.diagram.static_structure.ui.ClassDiagramRenderer org.argouml.ui.explorer.TransferableModelElement org.argouml.uml.diagram.sequence.ui.SequenceDiagramRenderer org.argouml.application.security.ArgoJarClassLoader org.argouml.uml.diagram.static_structure.layout. ClassDiagramModelElementFactory org.argouml.uml.cognitive.critics.CrMultipleAgg org.argouml.uml.cognitive.critics.CrMultiComposite org.argouml.uml.diagram.ui.SPFigEdgeModelElement org.argouml.uml.diagram.state.ui.StateDiagramRenderer org.argouml.uml.diagram.ui.ActionAddAssociationRole org.argouml.uml.diagram.deployment.ui.DeploymentDiagramRenderer org.argouml.ui.ActionGoToCritique org.argouml.application.security.ArgoAwtExceptionHandler org.argouml.uml.cognitive.critics.CrUnnavigableAssoc



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

**Logiciel analyse : DrJava****Version : v. 20030822**

Large classe	NAD	NMD	Les classes de donnees
edu.rice.es.drjava.ui. <b>MainFrame</b> (0.9944444441998547)	225.0	222.0	edu.rice.es.drjava.ui.BackgroundColorListener drjava-stable-20030822.src.edu.rice.es.drjava.model. definitions.reducedmodel.Stutter drjava-stable-20030822.src.edu.rice.es.drjava.model.definitions. reducedmodel.InsideDoubleQuote edu.rice.es.drjava.model.compiler.Javac141FromToolsJar edu.rice.es.drjava.model.compiler.JSR14v20FromSetLocation edu.rice.es.drjava.model.compiler.JSR14FromSetLocation edu.rice.es.drjava.model.compiler.JavacFromClasspath drjava-stable-20030822.src.edu.rice.es.drjava.model.definitions. reducedmodel.InsideSingleQuote drjava-stable-20030822.src.src-windows.edu.rice.es.drjava.platform. WindowsPlatform drjava-stable-20030822.src.src-mac14.edu.rice.es.drjava.platform. MacPlatform edu.rice.es.drjava.ui.ForegroundColorListener edu.rice.es.drjava.platform.PlatformFactory edu.rice.es.drjava.model.compiler.JSR14v12FromSetLocation drjava-stable-20030822.src.edu.rice.es.drjava.model.compiler. JavacFromToolsJar drjava-stable-20030822.src.edu.rice.es.drjava.model.compiler. GJv6FromClasspath edu.rice.es.drjava.model.compiler.Javac141FromClasspath drjava-stable-20030822.src.edu.rice.es.drjava.model.compiler. Javac141FromToolsJar edu.rice.es.drjava.model.compiler.JavacFromToolsJar drjava-stable-20030822.src.edu.rice.es.drjava.model.definitions. reducedmodel.Free edu.rice.es.drjava.model.compiler. GJv6FromClasspath drjava-stable-20030822.src.edu.rice.es.drjava.model.definitions. reducedmodel.InsideLineComment edu.rice.es.drjava.model.AlreadyOpenException drjava-stable-20030822.src.src-mac13.edu.rice.es. drjava.platform.MacPlatform



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

			<del>drjava-stable-20030822.src.edu.rice.cs.drjava.model.definitions.reducedmodel.IndentedBlockComment</del> <del>drjava-stable-20030822.src.edu.rice.cs.drjava.model.compiler.Javac141FromClasspath</del> <del>edu.rice.cs.drjava.model.FileMovedException</del> <del>drjava-stable-20030822.src.edu.rice.cs.drjava.model.compiler.JavacFromClasspath</del> <del>drjava-stable-20030822.src.edu.rice.cs.drjava.ui.ClasspathFilter</del> <del>drjava-stable-20030822.src.edu.rice.cs.drjava.platform.PlatformFactory</del>
edu.rice.cs.drjava.model.definitions. <b>IndentTest</b> (0.8920634911528655)	9.0	72.0	<del>drjava-stable-20030822.src.edu.rice.cs.drjava.model.definitions.reducedmodel.IndentInfo</del> edu.rice.cs.drjava.model.definitions.reducedmodel.IndentInfo
edu.rice.cs.drjava.model.repl.newjvm. <b>MainJVM</b> (0.938775505946607)	15.0	51.0	edu.rice.cs.drjava.model.compiler.JSR14v12FromSetLocation edu.rice.cs.drjava.model.compiler.JSR14FromSetLocation edu.rice.cs.drjava.model.compiler.JSR14v20FromSetLocation
edu.rice.cs.drjava.model. <b>DefaultGlobalModel</b> (0.963999999264876)	20.0	77.0	<del>drjava-stable-20030822.src.edu.rice.cs.drjava.model.definitions.reducedmodel.Stutter</del> <del>drjava-stable-20030822.src.edu.rice.cs.drjava.model.definitions.reducedmodel.IndentedDoubleQuote</del> <del>edu.rice.cs.drjava.model.compiler.Javac141FromToolsJar</del> <del>edu.rice.cs.drjava.model.compiler.JavacFromClasspath</del> <del>drjava-stable-20030822.src.edu.rice.cs.drjava.model.definitions.reducedmodel.IndentedSingleQuote</del> <del>drjava-stable-20030822.src.src-windows.edu.rice.cs.drjava.platform.WindowsPlatform</del> <del>drjava-stable-20030822.src.src-mac14.edu.rice.cs.drjava.platform.MacPlatform</del> <del>edu.rice.cs.drjava.platform.PlatformFactory</del> <del>drjava-stable-20030822.src.edu.rice.cs.drjava.model.compiler.JavacFromToolsJar</del> <del>drjava-stable-20030822.src.edu.rice.cs.drjava.model.compiler.GJv6FromClasspath</del> <del>edu.rice.cs.drjava.model.compiler.Javac141FromClasspath</del> <del>drjava-stable-20030822.src.edu.rice.cs.drjava.model.compiler.</del>



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

			<p>Javac141FromToolsJar  edu.rice.cs.drjava.model.compiler.JavacFromToolsJar  drjava-stable-20030822.src.edu.rice.cs.drjava.model.definitions.  reducedmodel.Free  edu.rice.cs.drjava.model.compiler.GJv6FromClasspath  drjava-stable-  20030822.src.edu.rice.cs.drjava.model.definitions.reducedmodel.  InsideLineComment  edu.rice.cs.drjava.model.AlreadyOpenException  drjava-stable-20030822.src.src-mac13.edu.rice.cs.drjava.platform.  MacPlatform  drjava-stable-  20030822.src.edu.rice.cs.drjava.model.definitions.reducedmodel.  InsideBlockComment  drjava-stable-20030822.src.edu.rice.cs.drjava.model.compiler.  Javac141FromClasspath  edu.rice.cs.drjava.model.FileMovedException  drjava-stable-  20030822.src.edu.rice.cs.drjava.model.compiler.JavacFromClasspath  drjava-stable-20030822.src.edu.rice.cs.drjava.ui.ClasspathFilter</p>
edu.rice.cs.drjava.model. definitions. <b>DefinitionsDocument</b> (0.9715549920116906)	28.0	116.0	drjava-stable-20030822.src.edu.rice.cs.drjava.model.FileMovedException edu.rice.cs.drjava.model.TestDocGetter edu.rice.cs.drjava.model.definitions.indent.ActionStartStmtOfBracePlus edu.rice.cs.drjava.model.definitions.indent.ActionBracePlus edu.rice.cs.drjava.model.definitions.indent.ActionStartPrevLinePlusMultiline edu.rice.cs.drjava.model.definitions.indent.ActionStartPrevLinePlusBackup edu.rice.cs.drjava.model.FileMovedException edu.rice.cs.drjava.model.definitions.indent.ActionStartPrevLinePlus edu.rice.cs.drjava.model.definitions.indent.ActionStartCurrStmtPlus
edu.rice.cs.drjava.ui. <b>DebugPanel</b> (0.9686520378190008)	29.0	46.0	drjava-stable- 20030822.src.edu.rice.cs.drjava.ui.config.VectorOptionComponent



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

**Logiciel analyse : DrJava****Version : v.20040326**

Large classe	NAD	NMD	Les classes de données
edu.rice.cs.drjava.model.definitions. DefinitionsDocument (1.010752688172043)	21.0	96.0	edu.rice.cs.drjava.model.definitions.indent. ActionStartPrevLinePlusMultiline edu.rice.cs.drjava.model.definitions.indent. ActionStartPrevLinePlusBackup edu.rice.cs.drjava.model. FileMovedException edu.rice.cs.drjava.model.definitions.indent. QuestionPrevLineStartsWith edu.rice.cs.drjava.model.definitions.indent. ActionStartPrevLinePlus edu.rice.cs.drjava.model.definitions.indent. ActionStartStmtOfBracePlus edu.rice.cs.drjava.model.definitions.indent. ActionBracePlus edu.rice.cs.drjava.model.definitions.indent. ActionStartCurrStmtPlus edu.rice.cs.drjava.model.definitions.indent. ActionStartPrevLinePlusMultilinePreserve



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

**Spaghetti-Code**

**Logiciel analyse : ArgoUml**

**Version : suggéré par l'encadrant**

Nom de la Classe	Valeur du CBO	Valeur du LCOM5	Méthodes sans paramètres
org.argouml.uml.diagram.ui. ImageSelection	2.0	1.5	getTransferDataFlavors
org.argouml.uiexplorer.rules. IncomingDependencyNode	5.0	1.5	getParent toString
org.argouml.uml.ui.behavior. state_machines.PropPanelState	2.0	1.1111111100763083	getDeferrableEventsScroll getDoScroll getEntryScroll getExitScroll getInternalTransitionsScroll
org.argouml.swingext. SpacerPanel	1.0	1.5	getMinimumSize getPreferredSize getSize
org.argouml.uml.diagram.sequence. ui.FigLinkPort	2.0	1.1666666567325592	getActivationStatus getGravityPoints
org.argouml.uml.diagram.sequence. LinkNode	2.0	1.0833333283662796	getFigLinkPort getObject isDestroyed
org.argouml.uml.ui. UMLMultiplicityComboBoxModel	3.0	1.25	buildModelList
org.argouml.util.CustomSeparator	5.0	1.1	getPeekCount hasFreePart reset tokenLength
org.argouml.application. helpers.DiagramHelper	1.0	1.1428571428571428	getDiagramMenuItem getModuleAuthor getModuleDescription getModuleKey



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

			getModuleName getModuleVersion initializeModule isModuleEnabled shutdownModule
org.argouml.uml.cognitive. ProjectMemberToDoList	4.0	1.3333333333333333	getResolvedCriticsList getToDoList getType getZipFileExtension
org.argouml.util. ExprSeparatorWithStrings	2.0	1.3333333333333333	hasFreePart reset tokenLength
org.argouml.ui. ShadowComboBox	2.0	1.0	access\$0
org.argouml.uml.ui.foundation. core.PropPanelNamespace	5.0	1.0	addClass addInterface addPackage getOwnedElementsScroll
org.argouml.util.LineSeparator	1.0	1.2	getPeekCount hasFreePart reset tokenLength
org.argouml.ui.explorer. rules.AssociationsNode	5.0	1.5	getParent toString
org.argouml.util. QuotedStringSeparator	3.0	1.3333333333333333	hasFreePart reset tokenLength
org.argouml.uml. reveng.classfile.ShortAST	3.0	1.0	getShortValue
org.argouml.uml.util. namespace.StringNamespaceElement	2.0	2.0	getNamespaceElement toString
org.argouml.language. java.generator.JavaUML	1.0	2.0	getCore
org.argouml.	2.0	1.5	getModuleAuthor



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

application.helpers.ModuleHelper			getModuleDescription getModuleKey getModuleName getModuleVersion initializeModule isModuleEnabled shutdownModule
org.argouml.uml. ui.foundation.core.PropPanelDependency	<b>5.0</b>	<b>1.0</b>	getClientScroll getSupplierScroll
org.argouml. ui.emd.ToDoItemAction	<b>1.0</b>	<b>1.0</b>	getRememberedTarget



**Détection semi-automatique des patrons de mauvaise conception  
dans les architectures orientées-objet**

---

**Logiciel analyse : GanttProject****Version : V1.10.2**

Nom de la Classe	Valeur du CBO	Valeur du LCOM5	Méthodes sans paramètres
net.sourceforge.ganttproject.resource.ResourceEvent	4.0	1.5	getManager getResource getResources
net.sourceforge.ganttproject.parser.ParsingContext	2.0	2.0	getTaskID
net.sourceforge.ganttproject.gui.projectwizard.ProjectNamePage	3.0	1.5	getComponent getTitle
net.sourceforge.ganttproject.shape.JPaintCombo	2.0	1.1666666666666667	addActionListener getSelectedIndex getSelectedItem getSelectedPaint setRenderer setSelectedIndex setSelectedItem
net.sourceforge.ganttproject.document.FileDocument	1.0	1.1428571428571428	canRead canWrite getDescription getFilePath getInputStream getOutputStream getPath isValidForMRU
net.sourceforge.ganttproject.task.algorithm.AlgorithmBase	1.0	2.0	isEnabled
net.sourceforge.ganttproject.task.dependency.SearchKey	2.0	1.5	hashCode
net.sourceforge.ganttproject.task.event.TaskScheduleEvent	0.0	1.0	getNewFinishDate getNewStartDate getOldFinishDate getOldStartDate
net.sourceforge.ganttproject.action.NewArtefactAction	4.0	1.5	getIconOnMouseOver putValue

