Playing Roles in Design Patterns : An Empirical Descriptive and Analytic Study

Foutse Khomh, Yann-Gaël Gueheneuc, and Giuliano Antoniol

Cédric Tessier

24 septembre 2021

Plan i

Plan ii

Concepts et définitions

Concepts et définitions

Design Pattern : Solution générale et réutilisable à un problème commun et redondant dans un contexte logiciel.

Empirical Study : Étude basée sur les observations et les mesures dans le but d'accroître les connaissances et la compréhension d'un domaine [1].

Change-prone : Classe qui à un certain moment dans le temps aura changé plus que les autres classes [2].

Concepts et définitions

Motif : Échantillon fiable de traits, d'actes, de tendances ou d'autres caractéristiques observables d'un patron de conception [3].

DeMIMA : Framework multicouche basé sur la programmation par contrainte qui permet d'identifier un motif DM dans un code source S[3].

Role in motif : Rôle que peut jouer une classe dans un motif d'un patron de conception.

Processus de détection des rôles (image provenant de [3]) :

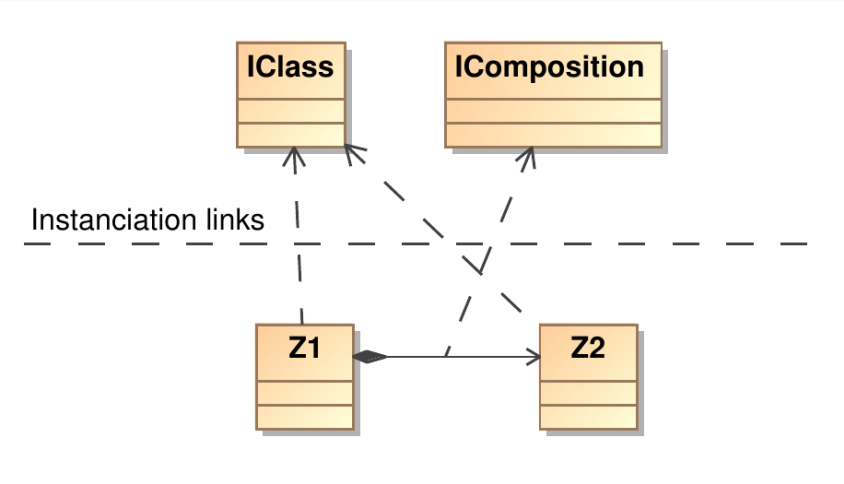

Figure 1 : Modèle du motif DM

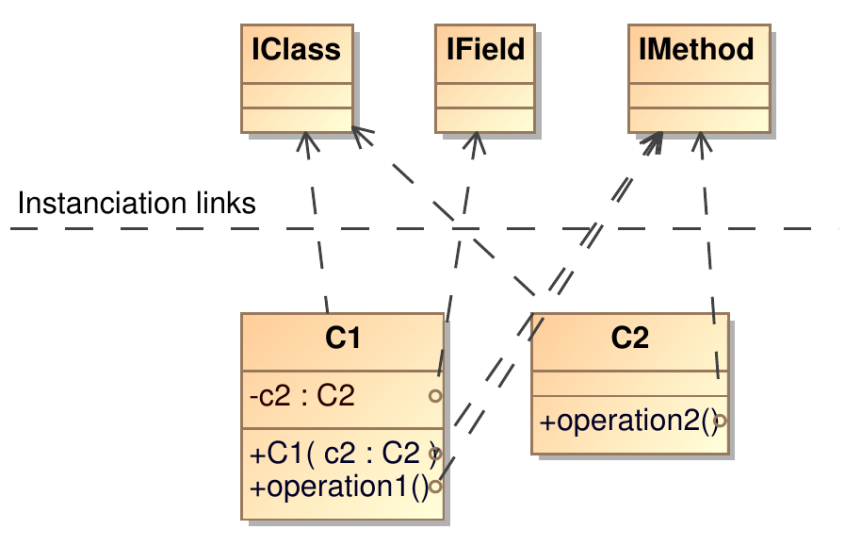

Figure 2 : Modèle du code S

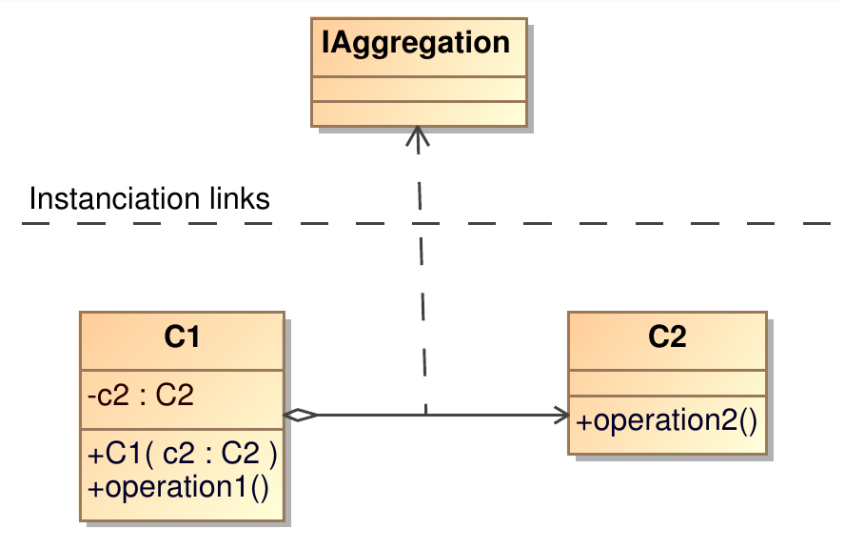

Figure 3 : Modèle du code S avec relation binaire

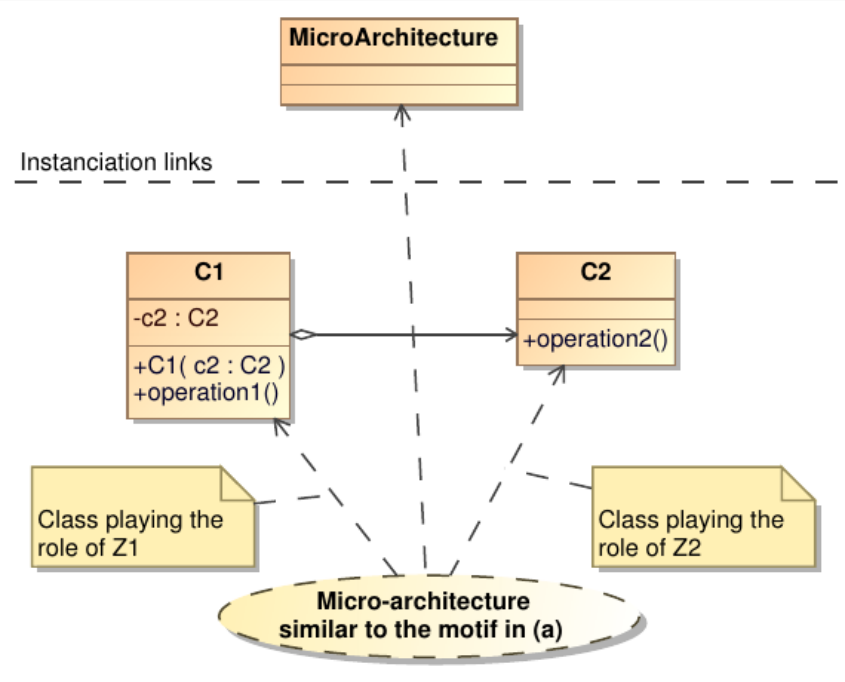

Figure 4 : Assignation des rôles du modèle DM similaire

Introduction

Motivations

- Une classe peut jouer n rôle dans m motifs ($\mathrm{n} > 0, \mathrm{m} > 0$).
- Les autres travaux ne tenaient pas compte de la différence entre $\mathrm{n} = 1$ et $\mathrm{n} > 1$

Travaux connexes

Que disent les autres travaux ?

- Dans une *Abstract Factory*, les classes concrètes changent plus souvent que les classes abstraites[4]
- Il existe une corrélation entre *Large Class* et les motifs *Observer*, *Singleton*[5]
- Les classes *Factory Method* sont plus compactes et moins couplées[5]

Protocole de l'étude

But

Étudier et mesurer l'impact pour une classe de jouer zéro, un ou deux rôles.

Questions de recherche

Descriptive :

RQ1 : Quelle est la proportion des classes jouant zéro, un ou deux rôles dans un nombre de motifs.

Analytiques :

RQ2 : Quelles sont les caractéristiques internes d'une classe qui sont le plus impactées en jouant un ou deux rôles comparativement à 0 rôle.

RQ3 : Quelles sont les caractéristiques externe d'une classe qui sont le plus impactées en jouant un ou deux rôles comparativement à 0 rôle.

Hypothèse nulle

$\mathrm{H}_{0\mathrm{mi/j}}$: La distribution des valeurs de la métrique m pour les classes jouant $\mathrm{i} \in [1, 2]$ rôles est similaire à celles des classes jouant $\mathrm{j} \in [0, 1] \wedge \mathrm{j} \neq \mathrm{i}$ rôle.

Variables

Indépendantes :

0-role Échantillon de classe jouant 0 rôle

1-role Échantillon de classe jouant 1 rôle

2-role Échantillon de classe jouant 2 rôles

Dépendantes :

Métriques Métriques pour les caractéristiques internes et externes

Caractéristiques

Internes (56 métriques différentes) :

CBO Coupling Between Objects

LCOM5 Cohesion in Methods

WMC Weighted Method Count

CC Cyclomatic Complexity

...

Externes :

Change-proneness Tendance d'une classe à changer

Analyse

Descriptive :

RQ1

1. Extraire les échantillons des classes provenant de 6 programmes à l'aide de l'approche DeMIMA.
2. Calculer la précision de DeMIMA avec une validation manuelle des classes jouant 1 et 2 rôles.
3. Extrapoler les proportions des échantillons dans la population générale.

Analytiques :

RQ2 et RQ3

1. Utiliser le test *Wilcoxon rank-sum* sur les paires d'échantillons pour valider $\mathrm{H}_{0\mathrm{mi/j}}$

Exécution de l'étude

Taille des échantillons

Hypothèses :

- Population suit une distribution normale

On détermine la taille des échantillons en utilisant les paramètres suivant :

- Seuil de signification : 0.05
- Effet de taille moyenne : 0.58
- Puissance du test : 0.8

Échantillons de 50 classes pour *t-test*

Taille des échantillons

On utilise le *Asymptotic Relative Efficiency* (ARE) pour comparer la puissance des tests pour chaque échantillon ce qui donne une puissance toujours supérieure à 0.864 (plus grand que le 80% du *t-tes*)

Pour être conservatif :

$$50/0.864 = 58$$

Échantillons de 58 classes pour *Wilcoxon test*

Choix échantillons

Les classes proviennent de six programmes Java de complexité, taille et maturité différentes :

- ArgoUML v0.18.1
- Azureus v2.1.0.0
- Eclipse JDT Core plug-in v2.1.2 (JDT Core v2.1.2)
- JHotDraw v5.4b2,
- Xalan v2.7.0
- Xerces v1.4.4

Choix des rôles

Les rôles principaux des motifs de conception suivants ont été utilisés dans l'étude :

Motif	Rôles
Command	Command, Invoker
Composite	Component, Composite
Decorator	Component, Decorator
Observer	Observer, Subject
Singleton	Singleton
State	Context, State

Évaluation des rôles

On utilise DeMIMA pour obtenir les classes jouant au moins 1 rôle. Un système de vote entre trois personnes permet par la suite de déterminer si la classe joue réellement 1 ou 2 rôles.

238 classes ont été validées manuellement pour estimer la précision de DeMIMA.

Calcul des variables dépendantes

Caractéristiques internes :

- POM frameword et PADL *meta-model*, *parser* ont été utilisés pour calculer les caractéristiques internes.

Caractéristiques externes :

- Les caractéristiques externes ont été calculées à l'aide du framework Ibdoos.

Résultats

Résultats (RQ1)

La proportion des classes jouant un ou deux rôles dans n'importe lequel des motifs varie de 4.02% à 30.72%.

Il est donc possible d'en conclure que les classes jouant un ou deux rôles existent dans les programmes et qu'elles sont non négligeables. Cela confirme le besoin d'en apprendre plus sur les classes jouant un certain nombre de rôles.

Résultats (RQ2)

Pour l'hypothèse $H_{0mi/j}$, seulement 8 métriques n'ont pas changé significativement entre les trois distributions : ANA, connectivity, CP, DSC, MFA, NOH, PP, et RPII.

- 29 métriques changent significativement entre les distributions zéro et un rôle.
- 48 métriques changent significativement entre les distributions zéro et deux rôles.
- 26 métriques changement significativement entre les distributions un et deux rôles.

Il y donc un impact significatif lorsqu'une classe joue deux rôles par rapport à un ou zéro rôle.

Résultats (RQ3)

Pour l'hypothèse $\mathrm{H}_{0\mathrm{mi/j}}$, seulement la distribution un par rapport à deux rôles n'a pas pu être rejeté.

- 56% des changements proviennent des classes jouant deux rôles
- 33% des changements proviennent des classes jouant un rôle.

Il y a donc un impact significatif lorsqu'une classe joue un ou plusieurs rôles sur la tendance à changer.

Menace à la validité

Menace à la validité

Construct Validity

- Les effets d'un rôle en particulier ont été ignorés
- Les classes jouant deux rôles dans le même motif ont été ignorées
- Limitation aux rôles principaux

Internal Validity

- Précision de DeMIMA

Menace à la validité

External Validity

- Projet open-source seulement
- Projet en Java seulement
- Limitation du nombre de patrons de conception

Statistical Validity

- Limitation du nombre de classes jouant un certain nombre de rôles dans certains projets

Conclusion

Conclusion

- En moyenne 8.24% des classes jouent un rôle dans les six programmes
- En moyenne 17.81% des classes jouent deux rôles dans les six programmes
- Le nombre de rôles joués à un impact significatif sur les métriques des classes
- Les classes jouant deux rôles (respectivement un) représentent 56% (respectivement 33%)

Références

Penn State.
Empirical Research in the Social Sciences and Education.
Psu.edu, 2019. url :
https://guides.libraries.psu.edu/emp.

Foutse Khomh, Yann-Gael Gueheneuc et
Giuliano Antoniol. “Playing roles in design patterns : An empirical descriptive and analytic study”. In :
2009 IEEE International Conference on Software Maintenance
(sept. 2009). doi : 10.1109/icsm.2009.5306327.

 Y.-G. Gueheneuc et G. Antoniol. “DeMIMA : A Multilayered Approach for Design Pattern Identification”. In : IEEE Transactions on Software Engineering 34 (sept. 2008), p. 667-684. doi : 10.1109/tse.2008.48. (Visité le 22/09/2021).

 Massimiliano Di Penta et al. “An empirical study of the relationships
between design pattern roles and class change proneness”. In : 2008 IEEE International Conference on Software Maintenance (sept. 2008). doi : 10.1109/icsm.2008.4658070. (Visité le 24/09/2021).


M. Vokac. “Defect frequency and design patterns : an empirical study of industrial code”. In : IEEE Transactions on Software Engineering 30 (déc. 2004), p. 904-917. doi : 10.1109/tse.2004.99. (Visité le 16/08/2020).