

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

Foutse Khomh, Yann-Gaël Guéhéneuc,
and Giuliano Antoniol

ICSM'09

2009/09/22



Context

- Design patterns are proven solutions to recurrent design problems
- Classes in a program may play one or more roles in one or more design patterns

Motivation

(1/2)

- Some authors like Bieman and Di Penta showed that classes playing some role(s) in one design motif are more complex and/or change-prone than classes playing no roles
- Previous works only considered classes playing no role or some roles in design motifs



Motivation

(2/2)

- Study the the impact on classes of playing one role in a motif or two roles in two different motifs wrt. playing zero role
- Show how some previous work on design patterns could benefit from this novel fine-grain perspective



Outline

- Study Definition
- Study Design
- Study Results
- Revisiting Previous Work
- Conclusion and Future Work



Study Definition

- **Goal:** study classes playing zero, one, or two roles in some design motifs.
- **Quality focus:** is that playing zero, one, or two roles impact differently classes.
- **The perspective:** is that both researchers and practitioners should be aware of the impact of playing roles on classes to make inform design and implementation choices.



Objects: Programs

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



Objects: Motifs and their Roles

Pattern	Main roles
Command	Command, Invoker
Composite	Component, Composite
Decorator	Component, Decorator
Observer	Observer, Subject
Singleton	Singleton
State	Context, State



Objects: Selection of Roles

- We study only main roles because they are most likely to impact classes
- They allow us to concentrate on a fewer number of roles during the manual validation



Objects: Selection of Pairs

- We exclude pairs with the same role because identical roles in different motifs have similar characteristics.
- We exclude pairs involving roles from the same motif because a class playing both the roles of Composite.Component and Composite.Composite are degenerated case
- Consequently, we retain 45 possible pairs



Study Design

■ Research questions

– Descriptive

- RQ1: What is the proportion of classes playing zero, one, or two roles in some motif(s)?

– Analytic

- RQ2: What are the internal characteristics of a class that are the most impacted by playing one or two roles wrt. zero role?
- RQ3: What are the external characteristics of a class that are the most impacted by playing one or two roles wrt. zero role?



Study Design

- Null hypotheses

- $H_{0mi/j}$: the distribution of the values of metric m for the classes playing $i \in [1, 2]$ role (s) is similar to that of classes playing $j \in [0, 1]$ and $j \neq i$ role



Descriptive Research Question (1/2)

■ Analysis method

- Selection of a population of programs
- Computation of classes playing zero, one, and two roles with our identification approach DeMIMA
- Computation of the accuracy of our approach through manual validation
- Estimation of the proportion of one and two roles

Descriptive Research Question (2/2)

■ Result of RQ1

- Classes playing one or two roles do exist in programs and are not negligible

Programs	Total	One Role	Two Roles
ArgoUML v0.18.1	1,267	51	316
	100%	4.02%	24.94%
Azureus v2.1.0.0	591	67	75
	100%	11.33%	12.69%
JDT Core v2.1.2	669	46	178
	100%	6.88%	26.60%
JHotDraw v5.4b2	413	24	101
	100%	5.81%	24.45%
Xalan v2.7.0	734	36	104
	100%	4.90%	14.16%
Xerces v1.4.4	306	94	56
	100%	30.72%	18.30%
Total	3,980	318	830
	100%	7.99%	20.85%

Analytic Research Question (1/12)

■ Analysis method

- We use the Wilcoxon rank-sum test
- We compute for each metric and each pair of samples (0-role, 1-role), (0-role, 2-roles), and (1-role, 2-role), the p-values for the corresponding null hypotheses



Analytic Research Question (2/12)

- Independent variables
 - Three samples of classes playing zero, one, and two roles in design motifs
 - We name these samples 0-role sample, 1-role sample, and 2-role sample
- BUT problem of the Oracle
 - Use of samples



Analytic Research Question (3/12)

■ Sample size

- We compute the sample size in two steps
 - First we assume the normality of the population and compute for a two-sample t-test
 - We adjust this size based on the Asymptotic Relative Efficiency (ARE)
- We obtain a sample size of 58

Analytic Research Question (4/12)

■ Sample size

- We distribute the sample size along the strata formed by the six programs

Programs	Expected	Zero Role	One Role	Two Roles
ArgoUML v0.18.1	17	17	10	10
Azureus v2.1.0.0	9	9	9	9
JDT Core v2.1.2	10	10	17	17
JHotDraw v5.4b2	6	6	6	6
Xalan v2.7.0	11	11	11	11
Xerces v1.4.4	5	5	5	5
Total	58	58	58	58

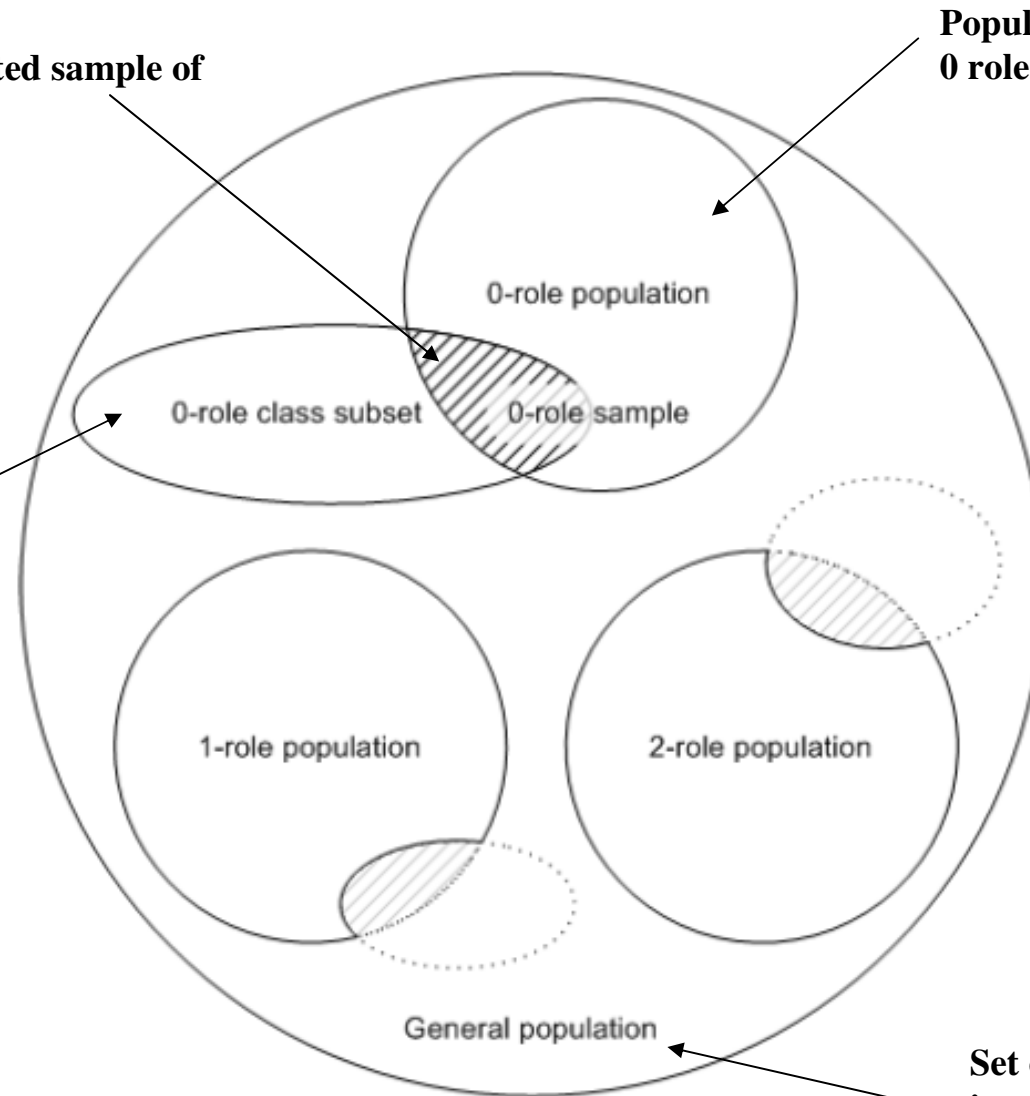
Analytic Research Question (5/12)

■ Building of the Samples

Manually validated sample of 0-role classes

Population of classes playing 0 roles in some design motifs

Subset of the classes in the general population that has been manually studied to identify 0-role classes



Set of all classes and interfaces belonging to the 6 programs



Analytic Research Question (6/12)

■ Building of the samples

- We manually validated 238 classes detected by DeMIMA as playing roles in a design pattern
- Each class was marked as true when it played one role (respectively, two roles) or false else
 - 88 classes played 1 role
 - 69 classes played 2 roles
 - 81 classes was playing no roles



Analytic Research Question (7/12)

■ Parameters of the study

- We obtained the following parameters for our study
 - Power: 0.8 (typical)
 - Significance level: 0.05 (typical)
 - Minimum effect size: 0.58 (medium)
 - Sizes of the samples: 58 classes



Analytic Research Question (8/12)

■ Dependent variables

- Internal Characteristics: 56 different metrics from the literature
 - Briand et al.'s class method import and export coupling
 - Chidamber and Kemerer's Coupling Between Objects (CBO), Lack of Cohesion in Methods (LCOM5), and Weighted Method Count (WMC)
 - Hitz and Montazeri 'C' connectivity of a class
 - Lorenz and Kidd numbers of new, inherited, and overridden methods and total number of methods.
 - McCabe's Cyclomatic Complexity Metric (CC)
 - Tegarden et al.'s numbers of hierarchical levels below a class and class-to-leaf depth



Analytic Research Question (9/12)

■ Dependent variables

- External Characteristics: Change-proneness of classes in this study
 - Computed the numbers and frequencies of past and future changes per class.

Analytic Research Question (10/12)

Results RQ2

Metric Groups	Metric Names	1 role vs. 0 role		2 role vs. 0 role		2 role vs. 1 role	
		<i>p</i> -values	Trends	<i>p</i> -values	Trends	<i>p</i> -values	Trends
Cohesion	CAM	0.854		0.0001996	↗	0.0003884	↗
	cohesionAttributes	0.6881		0.04051	↗	0.0009488	↗
	LCOM1	0.01313	↘	6.22E-09	↗	0.0009946	↗
	LCOM2	0.01087	↘	1.41E-07	↗	0.0017	↗
	LCOM5	0.03454	↗	3.95E-06	↗	0.001383	↗
Complexity	McCabe	0.2274		7.85E-07	↗	0.00063	↗
	SIX	0.004657	↗	1.41E-08	↗	0.0008183	↗
	WMC1	2.09E-05	↗	4.00E-08	↗	0.0467	↗
	WMC	0.01453	↘	5.40E-07	↗	0.001297	↗
Coupling	ACAIC	0.1733		0.03935	↗	0.5029	
	ACMIC	0.284		0.002702	↗	0.04961	↗
	CBO	0.5706		0.0001434	↗	0.001948	↗
	CBOin	0.191		7.89E-06	↗	0.0005939	↗
	CBOout	0.1055		5.96E-07	↗	0.0001025	↗
	connectivity	0.5005		0.07963		0.2603	
	CP	0.9802		0.2272		0.1428	
	DCAEC	9.37E-06	↗	0.003612	↗	0.06724	
	DCC	0.4149		2.98E-05	↗	0.002347	↗
	DCMEC	0.0001468	↗	0.001024	↗	0.595	
	PP	0.829		0.1382		0.1468	
	RFP	0.04845	↗	0.01477	↗	0.6074	
	RRFP	0.0968		0.02306	↘	0.5106	
	RRTP	0.02637	↘	0.03722	↘	0.6952	
RTP	0.2005		0.01295	↗	0.3693		

Not significant (8), Significant **29** **48** **26**

Analytic Research Question (11/12)

Results RQ2

Metric Groups	Metric Names	1 role vs. 0 role		2 role vs. 0 role		2 role vs. 1 role	
		<i>p</i> -values	Trends	<i>p</i> -values	Trends	<i>p</i> -values	Trends
Inheritance	AID	0.126		0.0001542	↗	0.1391	
	ANA	0.3958		0.8077		0.3918	
	CLD	< 2.2e-16	↗	7.94E-11	↗	0.003298	↘
	DIT	0.08713		8.59E-05	↗	0.2632	
	NCM	0.00087	↗	4.84E-09	↗	0.07486	
	NOC	2.22E-16	↗	3.55E-11	↗	0.245	
	NOD	2.22E-16	↗	5.29E-11	↗	0.07351	
	NOH	0.5644		0.601		0.9663	
	NOP	0.2248		6.10E-06	↗	0.007146	↗
	ICHClass	0.03035	↗	2.03E-07	↗	0.001095	↗
Polymorphism and Size	CIS	9.22E-07	↗	1.50E-08	↗	0.1605	
	DAM	0.1285		1.94E-05	↗	0.003362	↗
	DSC	0.1461		0.2098		0.8725	
	EIC	0.0002848	↗	9.03E-06	↗	0.5616	
	EIP	7.26E-13	↗	1.43E-09	↗	0.1039	
	MFA	0.1138		0.7105		0.243	
	MOA	0.0001883	↗	6.44E-10	↗	0.01493	↗
	NAD	0.1349		5.03E-06	↗	0.003884	↗
	NADExtended	0.1514		1.14E-05	↗	0.005466	↗
	NCP	5.39E-06	↗	0.01465	↗	0.1198	
	NMA	9.34E-06	↗	2.30E-06	↗	0.3157	
	NMD	2.09E-05	↗	4.00E-08	↗	0.0467	↗
	NMDExtended	3.37E-05	↗	1.07E-07	↗	0.05112	
	NMI	0.1029		0.0001075	↗	0.2016	
	NMO	0.00163	↗	3.57E-10	↗	0.0005408	↗
	NOA	0.1868		7.35E-08	↗	0.01153	↗
	NOM	2.09E-05	↗	4.00E-08	↗	0.0467	↗
	NOPParam	7.81E-06	↗	2.38E-08	↗	0.1551	
	NOPM	2.89E-14	↗	1.93E-10	↗	0.2793	
	PIIR	7.00E-05	↗	0.01216	↗	0.2846	
REIP	5.94E-10	↗	7.54E-08	↗	0.3336		
RPII	0.1486		0.08605		0.8614		

Expected more inheritance and connection

Shallow inheritance tree

Analytic Research Question (12/12)

■ Results RQ3

Metric Groups	Metric Names	1 role vs. 0 role		2 role vs. 0 role		2 role vs. 1 role	
		<i>p</i> -values	Trends	<i>p</i> -values	Trends	<i>p</i> -values	Trends
Changeability	Frequencies of Past Changes	8.26E-07	↗	1.24E-09	↗	0.08794	
	Frequencies of Future Changes	0.0001564	↗	7.44E-06	↗	0.5983	
	Numbers of Past Changes	3.54E-07	↗	5.50E-10	↗	0.06668	
	Numbers of Future Changes	0.001552	↗	9.72E-05	↗	0.7018	

- Playing roles do impact the number of changes as well as the frequencies of the changes.
- Confirming previous studies that playing roles has a major impact on change-proneness.
- However, there is no significance difference between playing one role and two roles for change-proneness.



Revisiting previous work

- Bieman and McNatt's Work. Our result confirm their claim that design motifs impact the cohesion and coupling of programs.
- Hannemann and Kiczales' Work
 - The decrease in cohesion and increase in coupling may suggest that design motif-related methods may be orthogonal to the responsibilities of the classes.
 - Confirming that design motifs are often “cross-cutting concern” that could benefit from being “separated” using, for example, aspect-oriented programming.

Revisiting previous work

- Di Penta et al.'s Work. We replicate their study and found that:
 - Their results on the change-proneness of classes playing roles in design motifs was in large part due to classes playing 2 roles.
 - Classes playing 2 roles change 1.52 times more than classes playing one role.
 - Developers should be careful with classes playing roles, in particular those playing two roles.

Conclusion

(1/2)

- We answered the research questions
 - RQ1. In average, 8.24% (respectively 17.81%) of the classes of the six studied programs played one role (respectively two roles) in some motifs.
 - RQ2. There is a significant increase in many metric values, in particular for classes playing two roles. Confirming a posteriori the warning addressed to the community by Bieman, Beck, and others on the use of design patterns.



Conclusion

(2/2)

- RQ3. There is a significant increase in the frequencies and numbers of changes of classes playing two roles.
- We confirmed our samples on previous results.
- We justify the usefulness of this study by revisiting previous work.



Future work

- We plan to replicate this study on
 - Other motifs and programs
 - Classes playing three roles and more to confirm its generalisability
 - Analyse the semantic of classes playing roles in motifs

Questions



Thank you for listening



Ranking Design Motif Occurrences

We get inspiration from previous works by Antoniol et al. [1], Gueheneuc et al. [11], and Jahnke et al. [15] to use the study results to rank occurrences.

- We assign to each class in a program its probability to play one or more roles in a design motif using its metrics values.
- From the probability of classes, we assign a probability to an occurrence.

$$p_O = \frac{\sum_{i=1}^n \alpha_i \times p_{C_i}}{\sum_{i=1}^n \alpha_i}$$



Ranking Design Motif Occurrences

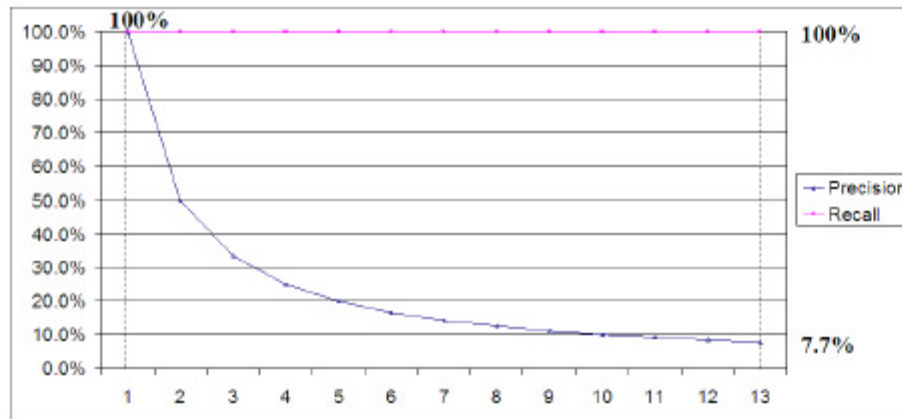
- We apply this naive approach using the metrics CAM, CBO, LCOM5, McCabe, MOA, NAD, NMO, SIX, and WMC.
 - ✓ This metrics have proven in this study to be the most discriminating of classes playing 0, 1, and 2.
 - ✓ We choose

$$\forall i \in [1, n], \alpha_i = 1$$

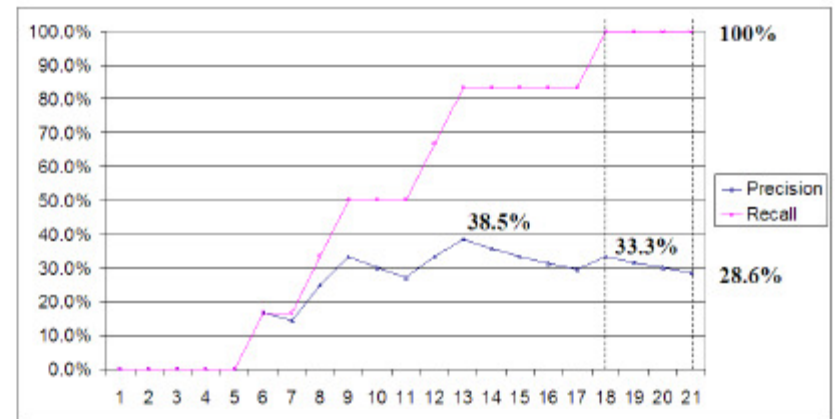
This approach on JHotDraw v5.1, and the occurrences of Decorator detected by DeMIMA leaded to 100% precision and recall on the first occurrence, to be compared to the previously reported 7.7% precision and 100% recall.

Ranking Design Motif Occurrences

Comparison with DeMIMA



(a) Precision and recall of the identification of Decorator.



(b) Precision and recall of the identification of State.

Conclusion

(2/2)

- We justify the usefulness of this study by revisiting previous work
- We propose a naive approach to rank occurrences which led to an increase in precision and recall
- Extending on this naïve approach, a new family of design pattern identification approaches could be designed to include the knowledge of the numbers of roles played by classes



Future work

- We plan to replicate this study on other motifs and programs as well as study classes playing three roles and more to confirm its generalisability
- We also plan to further study the ranking of occurrences using
 - Other a more sophisticated approach,
 - Other identification approaches,
 - Other programs,
 - Also, the use of Bayesian beliefs networks to assign probabilities