

IDS: An  
Immune-inspired  
Approach for the  
Detection of  
Software Design  
Smells

Salima Hassaine,  
Foutse Khomh,  
Yann-Gaël  
Guéhéneuc, and  
Sylvie Hamel

# IDS: An Immune-inspired Approach for the Detection of Software Design Smells

*System?*

Salima Hassaine, Foutse Khomh, Yann-Gaël Guéhéneuc,  
and Sylvie Hamel

Department of Computer and Software Engineering  
Université de Montréal, Québec, Canada  
École Polytechnique de Montréal, Québec, Canada

October 1, 2010

Context and  
Motivations

Limitations of  
Previous Approaches

Proposed  
Approach

Experiments

Conclusion and  
Future Work

Question?

# Outline

Context and Motivations

Limitations of ~~previously~~ ~~EA~~ approaches

~~The~~ Proposed Approach

Experiments

Conclusion and Future ~~work~~

Question ~~?~~

# Context and Motivations

- ▶ **Design smells are poor solutions to recurring design problems**
  - ▶ Introduced by developers when designing and implementing classes
  - ▶ Negatively impact the quality of a system
- ▶ **Design smells identification is a crucial activity in software maintenance**
  - ▶ Important for researchers as well as for practitioners
- ▶ **Various approaches to detect ~~them~~ *smells* have been proposed**
  - ▶ ~~The D~~ design smells identification exploits quality metrics
  - ▶ Manual detection, ~~R~~ Rules cards (DECOR), Bayesian Beliefs Networks (BBNs), B-Splines (ABS)

# Limitations of ~~P~~reviously ~~A~~pproaches

## ► DECOR: true/false identification based on fixed threshold *S*

- "Yes, it is an antipattern — No, it is not an antipattern"
- Cannot report accurate information for borderline classes

*shift right* → Submarine effect: several classes may be very close to be identified as antipatterns but remain under the threshold *S* during their evolution

- Requires experts' knowledge and interpretation to define the rule cards

```
RULE_CARD : SpaghettiCode {  
  
    ...  
  
    RULE: LongMethodMethodNoParameter { INTER LongMethod MethodNoParameter } ;  
    RULE: LongMethod      { METRIC LOC_METHOD VERY_HIGH 10.0 } ;  
    RULE: MethodNoParameter { METRIC MNOPARAM VERY_HIGH 10.0 } ;  
  
    ...  
  
    RULE: NoInheritance      { METRIC DIT 1 0.0 } ;  
    RULE: NoPolymorphism     { STRUCT NO_POLYMORPHISM } ;  
  
    RULE: ProceduralNameGlobalVariable { UNION ProceduralName GlobalVariable } ;  
    RULE: ProceduralName      { LEXIC CLASS_NAME (Make, Create, System, Exec) } ;  
    RULE: UseGlobalVariable   { STRUCT USE_GLOBAL_VARIABLE } ;  
  
} ;
```

# Limitations of ~~previously~~ ~~A~~ approaches

- ▶ **DECOR: true/false identification based on fixed threshold**
  - ▶ Yes, it is an antipattern - No, it is not an antipattern
  - ▶ Cannot report accurate information for borderline classes
  - ▶ Submarine effect: several classes may be very close to be identified as antipatterns but remain under the threshold during their evolution
  - ▶ Requires experts' knowledge and interpretation to define the rule cards

```
RULE_CARD : SpaghettiCode {  
    ...  
    RULE: LongMethodMethodNoParameter { INTER LongMethod MethodNoParameter } ;  
    RULE: LongMethod { METRIC LOC_METHOD VERY_HIGH 50.0 } ;  
    RULE: MethodNoParameter { METRIC MNNOPARAM VERY_HIGH 10.0 } ;  
    ...  
    RULE: NoInheritance { METRIC DIT 1 0.0 } ;  
    RULE: NoPolymorphism { STRUCT NO_POLYMORPHISM } ;  
    ...  
    RULE: ProceduralNameGlobalVariable { UNION ProceduralName GlobalVariable } ;  
    RULE: ProceduralName { LEXIC CLASS_NAME (Make, Create, System, Exec) } ;  
    RULE: UseGlobalVariable { STRUCT USE_GLOBAL_VARIABLE } ;  
} ;
```

# Limitations of ~~Previous~~ Approaches

IDS: An Immune-inspired Approach for the Detection of Software Design Smells

Salima Hassaine, Foutse Khomh, Yann-Gaël Guéhéneuc, and Sylvie Hamel

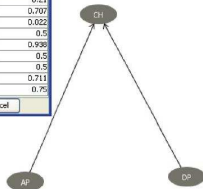
- ▶ **BBN: returns the probabilities of classes to be ~~smells~~ antipatterns but**

- ▶ Input ~~N~~ Nodes: characterizations of the design of a class,
- ▶ Output ~~N~~ Nodes: probability that the class is an antipattern
- ▶ Requires experts' knowledge to define a learning structure
- ▶ Incomplete experts' knowledge can cause a high number of false positives

Probability Distribution Table For CH

AP	DP	NO	YES
no smell	one role	0.79	0.21
no smell	more role	0.293	0.707
no smell	no role	0.978	0.022
more smell	one role	0.5	0.5
more smell	more role	0.062	0.938
more smell	no role	0.5	0.5
one smell	one role	0.5	0.5
one smell	more role	0.289	0.711
one smell	no role	0.25	0.75

Randomize Ok Cancel



Context and Motivations

Limitations of previously approaches

The Proposed Approach

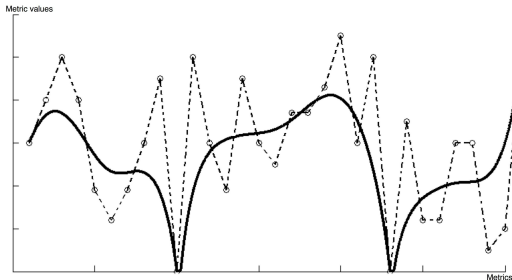
Experiments

Conclusion and Future work

Question ?

## Limitations of previously approaches

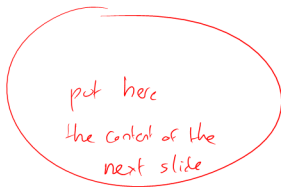
- returns the probabilities of classes to be smells*
- ▶ ~~ABS: Anticipation identification using B-Splines~~
    - ▶ Uses B-splines to abstract the signature of classes
    - ▶ A class is modelled using specific interpolation curves (i.e., B-splines) of plots mapping metrics and their values for the class
    - ▶ Focuses on detecting one kind of design smells at a time, while some smells are similar and classes with characteristics similar but no identical to some smells are also of interest to developers



# The Proposed Approach



## IDS: An Immune-inspired Approach for the Detection of Software Design Smells



⇒ give immediately an idea of what we do

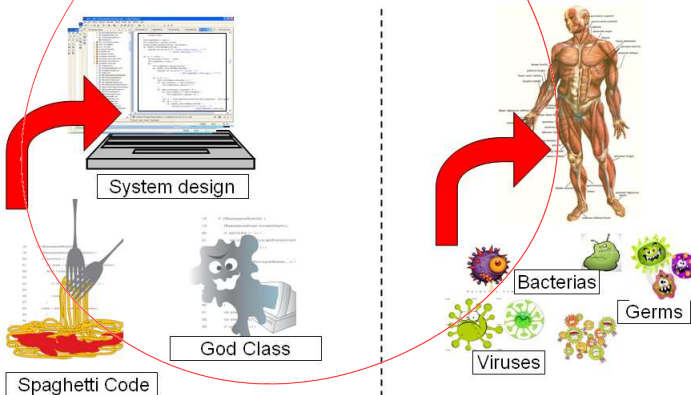
*become a "bullet point in the previous slide"*

# Parallel between object-oriented software systems and living bodies

IDS: An Immune-inspired Approach for the Detection of Software Design Smells

Salima Hassaine, Foutse Khomh, Yann-Gaël Guéhéneuc, and Sylvie Hamel

*merge with previous slide*



*(very nice!)*

# Parallel between object-oriented software systems and living bodies

Proposed Approach

Concepts of Immune System	
In Software	In Biology
System design	Body
Design smell detection approach	Immune system
Well-designed classes	Self-Cells
Smelly classes	Non-Self Cells
Sequence of metrics values	Antigen
Known pattern of metrics values (the learning phase)	Antibody
Similarity measure	Affinity

Table: Instantiation of an AIS to detect design smells.

remove or use as bullet point

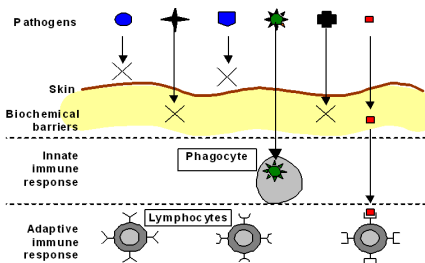
# Basic Immunology

*Proposed Approach*

## Role of the Immune System:

- ▶ **Protect our bodies from infection**
  - ▶ Diseases, virus, bacteria, etc.
- ▶ **Primary immune response:** *against*
  - ▶ Launch a response to ~~invading~~ pathogens
- ▶ **Secondary immune response:**
  - ▶ Remember past encounters
  - ▶ Faster response the second time around

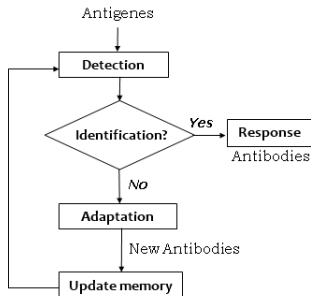
## How does it work?



# (AIS-based classification algorithm) Proposed Approach

## The Algorithm Principle :

- 1: Produce a large number of randomly-created *detectors* to recognise ~~the~~ antigens
- 2: Apply a *negative selection* mechanism to eliminate detectors that match the body's own cells
- 3: Detectors that match a pathogen are quickly multiplied via the *clonal selection* to accelerate the response to further attacks.



# Why IDS ?

Proposed Approach

## ► IDS vs. DECOR

- IDS does not use fixed thresholds
- IDS does not suffer ~~of~~ from the submarine effect

(very good!)

## ► IDS vs. BBNs

- IDS does not need experts' knowledge to define a learning structure
- IDS reduces the bias introduced in BBNs by the experts' subjectivity when structuring the BBNs ~~of the design smells~~

## ► IDS vs. ABS

- IDS doesn't focus on detecting one smell at a time.

## ► Discussion

- Several smells have similarities. For example, the Blob and Spaghetti Code antipatterns
- Classes similar but not equal to some design smell are also of interest to developers
- Previous approaches have somewhat limited performances in time, precision, and recall

# (Study Definition and Design)

Experiments

- ▶ IDS was used to identify three design smells (Blob, Functionnal Decomposition, Spaguetti Code)
- ▶ Accuracy of IDS was compared with DECOR and BBNs
- ▶ Case study conducted on two medium size open-source Java systems (GanttProject v1.10.2 and Xerces v2.7.0)
- ▶ Two experimental scenarios:
  - ▶ **Intra-system identification:** where historical data (i.e., correctly identified Blobs) are available for a given system and this data are used to identify other Blobs in the same system
  - ▶ **Extra-system identification:** where a quality analyst has access to the historical data from one system and uses it to identify design smells in the other system

# Study Definition and Design Experiments

Context and  
Motivations

Limitations of  
previously  
approaches

The Proposed  
Approach

Experiments

Conclusion and  
Future work

Question ?

We want to answer the two research questions:

- ▶ **RQ1:** To what extent an AIS-based approach can detect design smells in a system?
- ▶ **RQ2:** Is our approach better than state-of-the-art approaches, such as DECOR~~1~~ and BBNs?

# Study Definition and Design Experiments

- ▶ We answer **RQ1** in the following two scenarios:
  - ▶ **Intra-system identification:** We divide the classes of Xerces in three subsets with 16 occurrences of Blobs in each subset. Then, we train IDS on two of the subsets and apply it on the third subset (of the same system) in a 3-fold cross-validation.
  - ▶ **Extra-system identification:** We train IDS using the <sup>in</sup> design smells GanttProject and apply it to detect occurrences of design smells in the other system, Xerces. We also perform the same study in the other direction
- ▶ We answer **RQ2** by comparing the performance of IDS in precision, and recall, as computed in Scenario 2, against that of ~~previous approaches~~ (DECOR and BBNs).

# Study Results, Analyses and Discussions

## Answer to RQ1:

	Numbers of		Precisions	Recalls
	Design Smells	False Positives		
Subset 1	16	1	94.11%	100%
Subset 2	16	2	88.23%	100%
Subset 3	16	2	88.23%	100%
Average			90.19%	100%

Table: Intra-system detection on Xerces: 3-fold cross validation.

	GanttProject	Xerces
# of Classes	188	589
# of Design Smells Instances	20	54
# of False Positives	7	10
Precision	65.0%	81.48%
Recall	100%	100%

Table: Extra-system detection.

add one more section?

remove or use as bullet points

# Study Results, Analyses and Discussions

## Answer to RQ2:

Salima Hassaine,  
Foutse Khomh,  
Yann-Gaël  
Guéhéneuc, and  
Sylvie Hamel

	GanttProject			
	DECOR	BBNs	IDS	
			(Trained on Blob)	(Trained on Blob, FD, SC)
Blob	16 (8.5%)	7 (3.7%)	34	20 (10.6) 13 (6.9%) 65%
	4 (2.1%)	4 (2.1%)	4	
	25 %	57.1 %	11.76%	
FD	15 (8.0%)	-	-	
	4 (2.1%)	-	-	
	26.7%	-	-	
SC	14 (7.4%)	-	-	
	4 (2.1%)	-	11	
	28.5%	-	32.35%	
Average	26.73%	57.1 %		65%

Context and Motivations

Limitations of previously approaches

The Proposed Approach

Experiments

Conclusion and Future work

Question ?

	Xerces			
	DECOR	BBNs	IDS	
			(Trained on Blob)	(Trained on Blob, FD, SC)
	44 (7.6%)	41 (6.9%)	55	54 (9.1%) 44 (7.4%) 81.48%
	15 (2.5%)	15 (2.5%)	15	
	33.3%	36.5 %	27.27%	
	29 (5.6%)	-	-	
	15 (2.9%)	-	-	
	51.7%	-	-	
	76 (14.8%)	-	-	
	18 (3.0%)	-	18	
	23.68%	-	32.72%	
	36.22%	36.5%		81.48%

# Conclusion and Future work

- ▶ A novel approach (IDS) is ~~propose~~ <sup>work</sup> to identify occurrences of design smells

↳ The approach uses AIS-based classification to detect occurrences of design smells

↳ IDS outperforms both BBNs and DECOR (Generalisation, Parameter Stability, Adaptability, Portability across Systems, Simplicity and Self-regulatory, Performance) ✓

- ▶ **Future work** ~~work~~ <sup>bold</sup>
  - ▶ Replicating the evaluation of IDS on larger systems
  - ▶ Compare our approach with other machine learning techniques

# Question?

IDS: An  
Immune-inspired  
Approach for the  
Detection of  
Software Design  
Smells

Salima Hassaine,  
Foutse Khomh,  
Yann-Gaël  
Guéhéneuc, and  
Sylvie Hamel

Thanks for your attention



Context and  
Motivations

Limitations of  
previously  
approaches

The Proposed  
Approach

Experiments

Conclusion and  
Future work

Question ?