

# Un méta-modèle pour coupler application et détection des design patterns

Hervé Albin-Amiot

Pierre Cointe

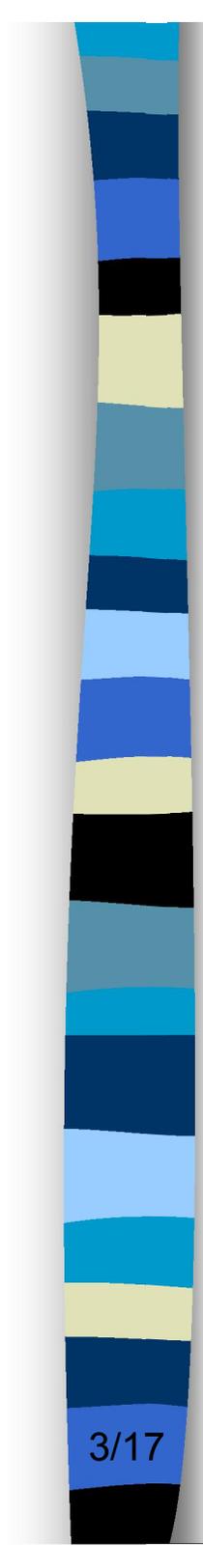
Yann-Gaël Guéhéneuc

{albin, cointe, guehene}@emn.fr



# Plan

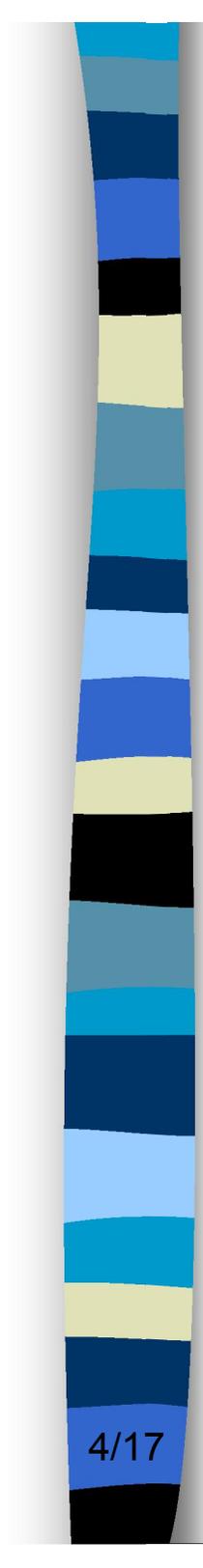
- Le contexte
- Notre objectif
- Nos besoins
- Travaux connexes
- Notre solution : le méta-modèle PDL
- PDL : principe & mise en œuvre
- Questions ouvertes



# Le contexte

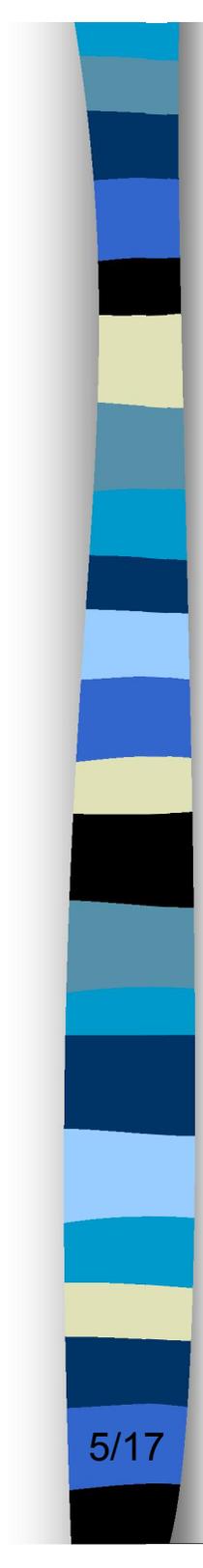
Améliorer la qualité des développements par objets en favorisant l'utilisation des **design patterns** pour aider à :

- Implémenter
- Comprendre
- Documenter



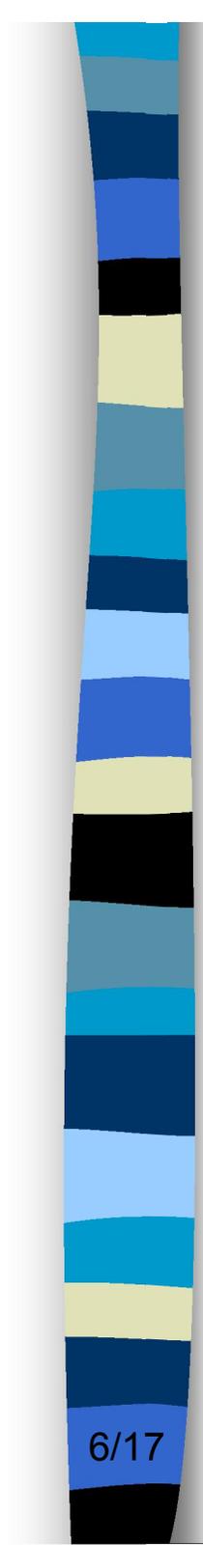
# Notre objectif

- Fournir un assistant logiciel aidant à :
    - L'implémentation
      - **Sélection**
      - Paramétrage
      - Génération
- } **Application**
- La compréhension et la documentation :  
détection de **formes exactes**



# Nos besoins

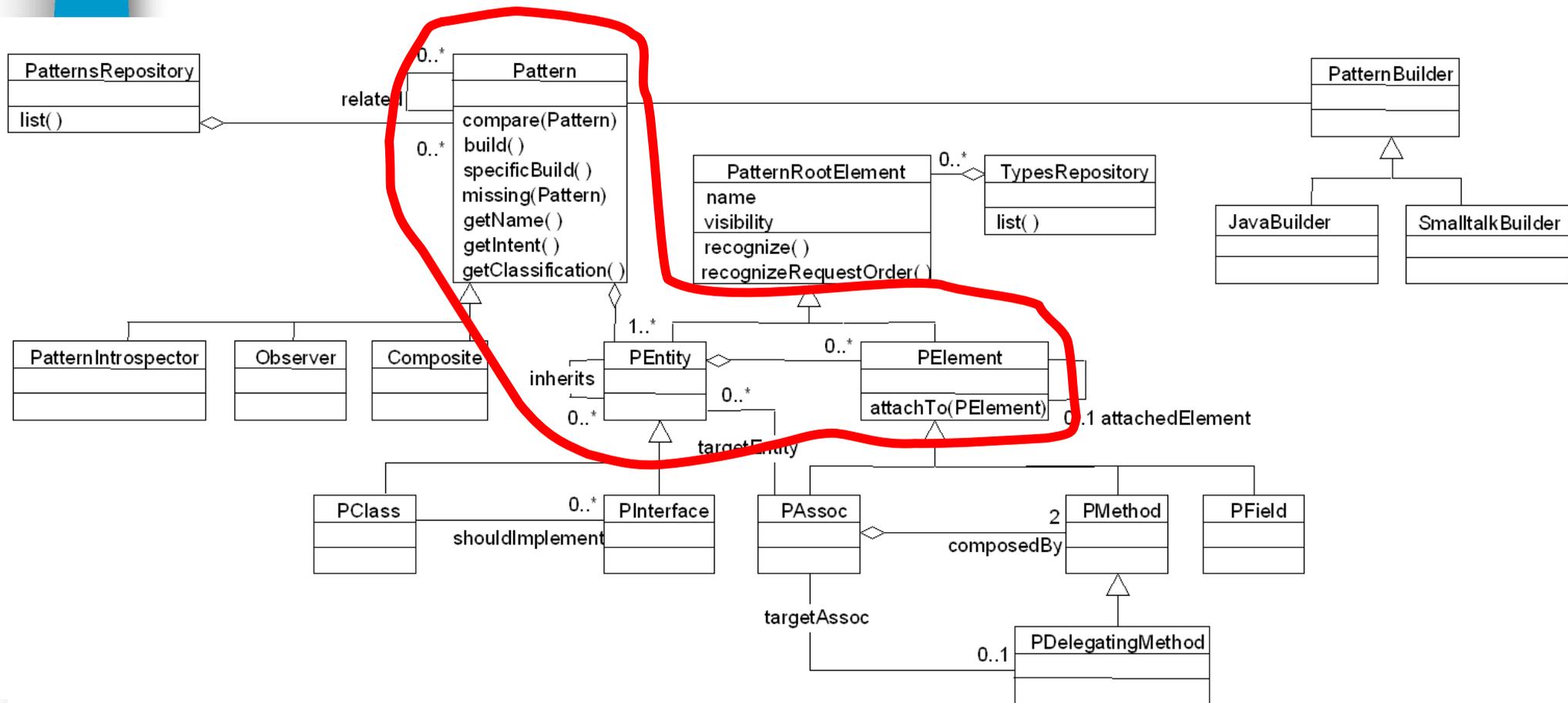
- Formaliser les design patterns
- Décrire les design patterns en tant que :
  - Entités de « **première classe** »
    - Elles savent comment produire du code
    - Elles savent identifier leurs propres occurrences dans du code
  - Entités **manipulables**
    - Sur lesquelles on peut raisonner
    - Que l'on peut adapter à un contexte spécifique
  - Entités avec « **traces** »



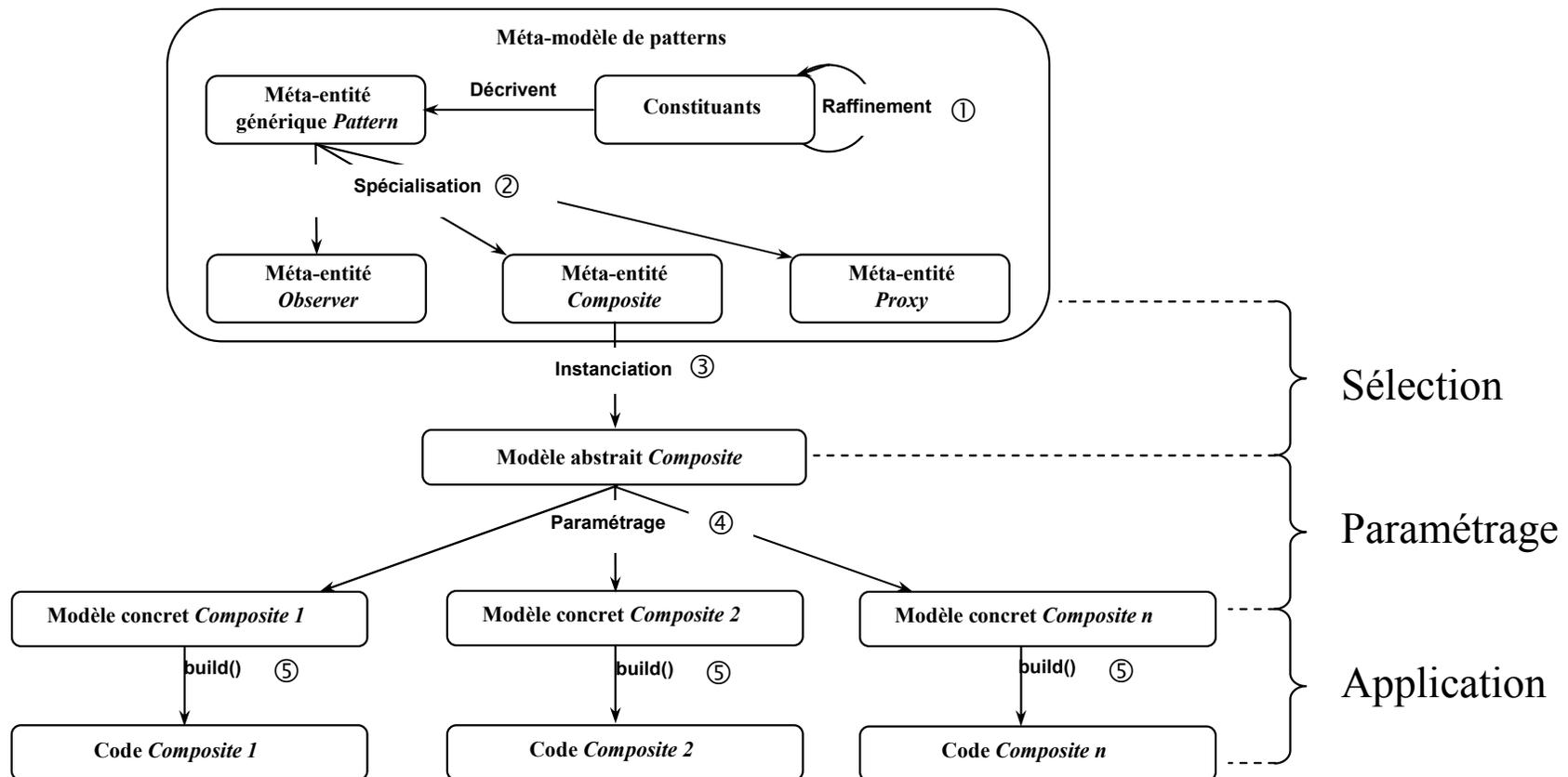
# Travaux connexes

- Formalisation :
  - LePUS [Eden 2000]
  - Fragment Model [Florjin *et al.* 1997]
- Application
  - [Budinsky *et al.* 1996]
- Détection
  - [Wuyts 1998], [Antoniol *et al.* 1998]

# Notre solution : le méta-modèle PDL

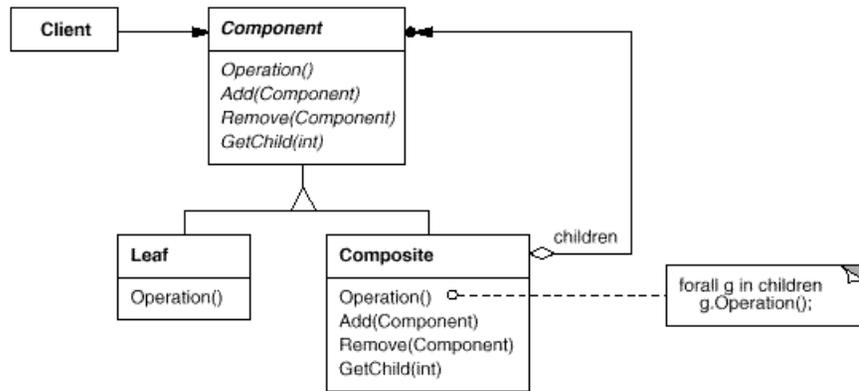


# PDL : principe



# PDL : mise en œuvre

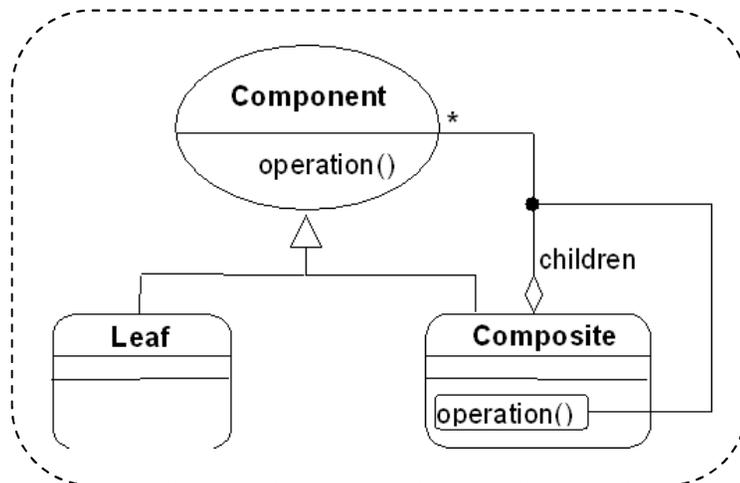
## Description



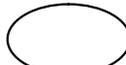
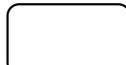
Descriptions informelles du [GoF]



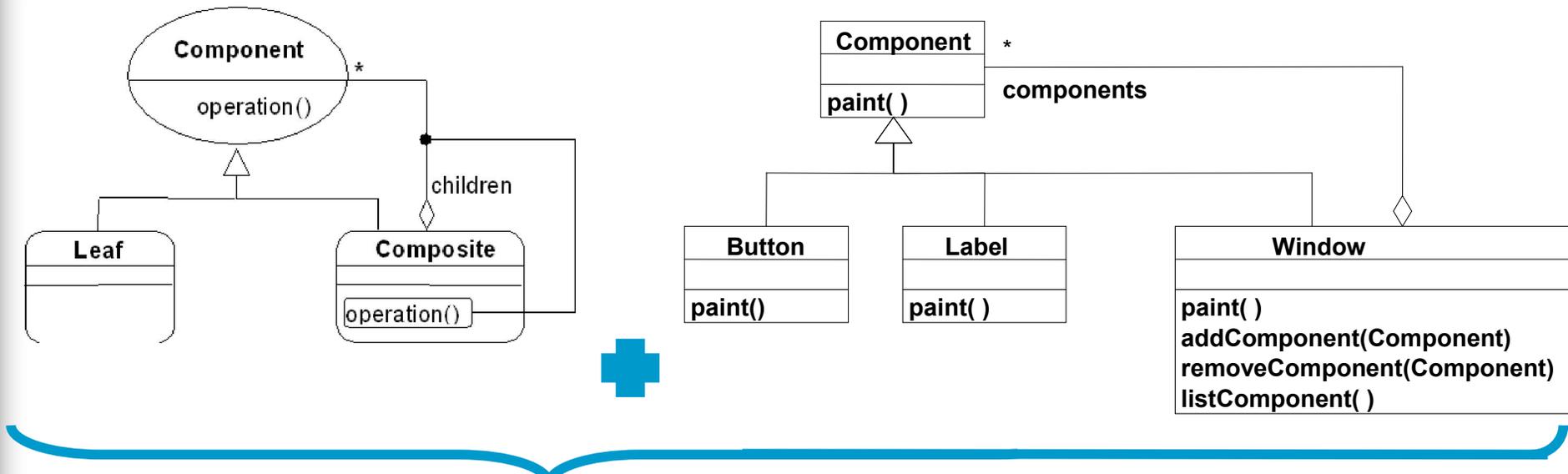
Traduction en un modèle de pattern



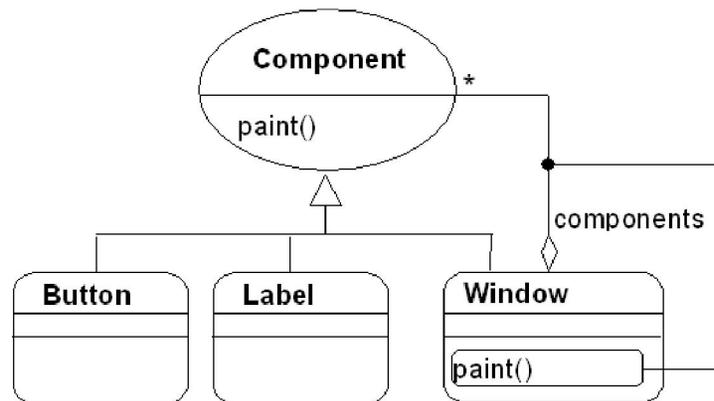
### Légende

-  Instance de Pattern
-  Instance de PInterface
-  Instance de PClass
-  Instance de PAssoc
-  Instance de PDelegation
-  Instance de PMethod

# PDL : mise en œuvre Application



Paramétrage



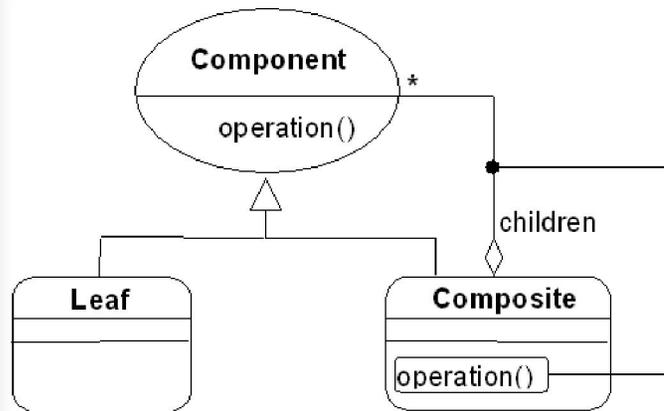
Génération



Code source  
Java, Claire

# PDL : mise en œuvre

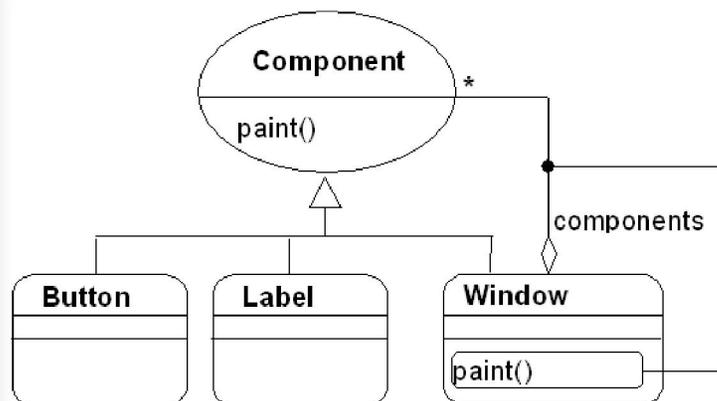
## Détection



Design pattern  
détecté ?



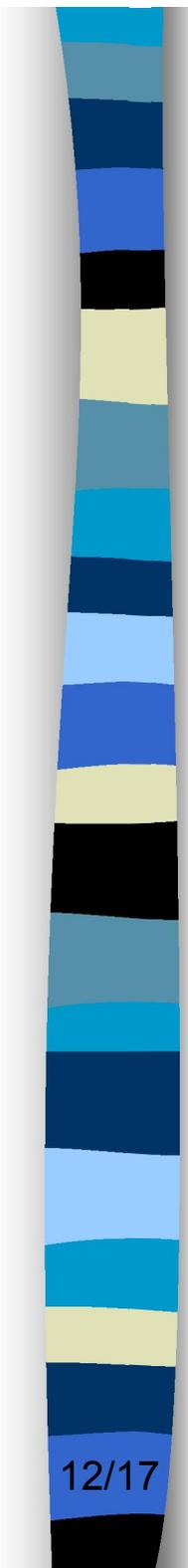
Comparaison  
(arc consistance + algorithmes dédiés)



Reconstruction



Code source Java

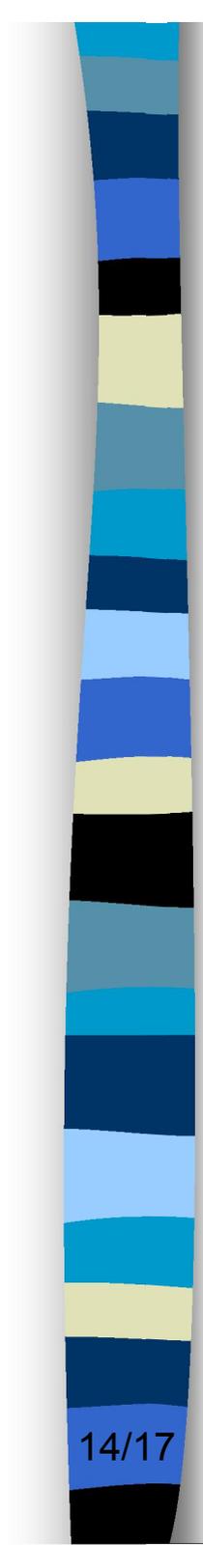


# Quelques résultats expérimentaux

- Application
  - Ptidej
- Détection sur plusieurs frameworks :
  - java.awt.\*
  - JHotDraw
  - JUnit
  - PatternsBox
  - Ptidej

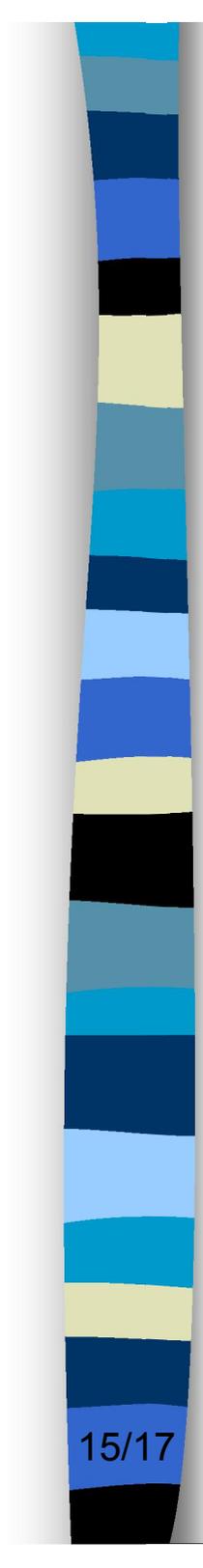
# Détection

Design patterns	Frameworks	NOC	Existing	Détection dans PatternsBox			
				Hits	Missed	False hits	Time (sec.)
Composite	java.awt.*	121	1	1			1
	JHotDraw	155	1	1			0,3
	JUnit	34	1		1		0,4
	JEdit	248					1,4
	PatternsBox	52					0,3
Decorator	java.awt.*	121					0,2
	JHotDraw	155	1	3		2	0,4
	JUnit	34	1	1			0,2
	JEdit	248					0,4
	PatternsBox	52					0,3
Factory Method	java.awt.*	121	3				1,1
	JHotDraw	155	2	2	1	1	0,5
	JUnit	34					
	JEdit	248					0,1
	PatternsBox	52					0,2
Iterator	java.awt.*	121					
	JHotDraw	155	3	3			0,1
	JUnit	34					
	JEdit	248	1	1			0,1
	PatternsBox	52					
Observer	java.awt.*	121					3,4
	JHotDraw	155	2	2			3,2
	JUnit	34	4	4			2,5
	JEdit	248	3	3			13
	PatternsBox	52	1	1			2,8
Singleton	java.awt.*	121	3	3			0,7
	JHotDraw	155	2	2			0,5
	JUnit	34					0,4
	JEdit	248					1
	PatternsBox	52	1	1			0,5
		<b>610</b>	<b>29</b>	<b>28</b>	<b>2</b>	<b>3</b>	<b>5,8</b>



# Réalisation de l'objectif

- Décrire, appliquer et détecter les design patterns
  - Entités de première classe manipulables
- Nous avons un assistant aidant :
  - L'implémentation
  - La détection et la documentation de formes exactes de design patterns(Cet assistant est « relativement » simple)



# Questions ouvertes

1/3

- Par rapport au contexte :

Améliorer la qualité des développements par objets en favorisant l'utilisation des design patterns

# Questions ouvertes

2/3

## ■ Formalisation :

- Un unique méta-modèle est-il suffisant ?
- Comment représenter les compromis ?

## ■ Application :

- Comment bien décrire le code source ?
- Est-ce qu'elle est toujours possible et intéressante ?

# Questions ouvertes

3/3

## ■ Compréhension :

- Est-il possible de détecter les formes exactes de n'importe quel design pattern ?
- Est-il possible de détecter les formes approchées de n'importe quel design pattern ?