

Dynamic analysis for Design Pattern detection in Java: Information collected using JPDA

Fabrizio Perin

Milano Bicocca University

March 10, 2009

Outline

- 1 Objective and Specifications
- 2 Rules
- 3 JADEPT architecture
- 4 Results
- 5 Conclusions
- 6 Future Development

1 Objective and Specifications

2 Rules

3 JADEPT architecture

4 Results

5 Conclusions

6 Future Development

Objective and Specifications

Objective

- Design and implementation of a prototype for design pattern detection for Java codes

Specifications

- Dynamic analysis: data collection through JPDA (Java Platform Debugger Architecture)
- Rules based approach
- Designed for *behavioral* design pattern: implemented for *Chain Of Responsibility (COR)*, *Observer* and *Visitor*

⇒ **J**ava **D**esign **P**attern de**T**ector

① Objective and Specifications

② Rules

③ JADEPT architecture

④ Results

⑤ Conclusions

⑥ Future Development

Rules: a definition

- A **rule** is a set of one or more conditions verified by a given design pattern
- Affinities of rules to different patterns are defined via specific weights
- Rules were based on definitions given by E. Gamma, R. Helm, R. Johnson, J. Vlissides (GoF) [1] and J. W. Cooper [2]
- Rules describe both structural and behavioral properties of selected patterns.
- Hierarchical relationships inter-connect rules
- Sets of rules with high specificity define *macrorules*

Rules: an example

Nr.	Rule	Weight/ Specificity	Type	Dependency
1	Some classes implement the same interface.	1	S	
2	Some classes extend the same class.	1	S	
3	All classes that implements the same interface or extend the same class, contain a reference whose type is the same of the implemented interface or the extended class.	3	S	
4	Each class has one method that contains a call to the same method in another class of the same family and this method must contain a parameter	3	S-D	
5	“handle” is define as the name of the method identified by the forth rule. The call to handle method of an object is always originated by the same caler object. This property is true for each object of the family	3?	D	If 4 = +1
M	Sequential redirection	4, 5		

1 Objective and Specifications

2 Rules

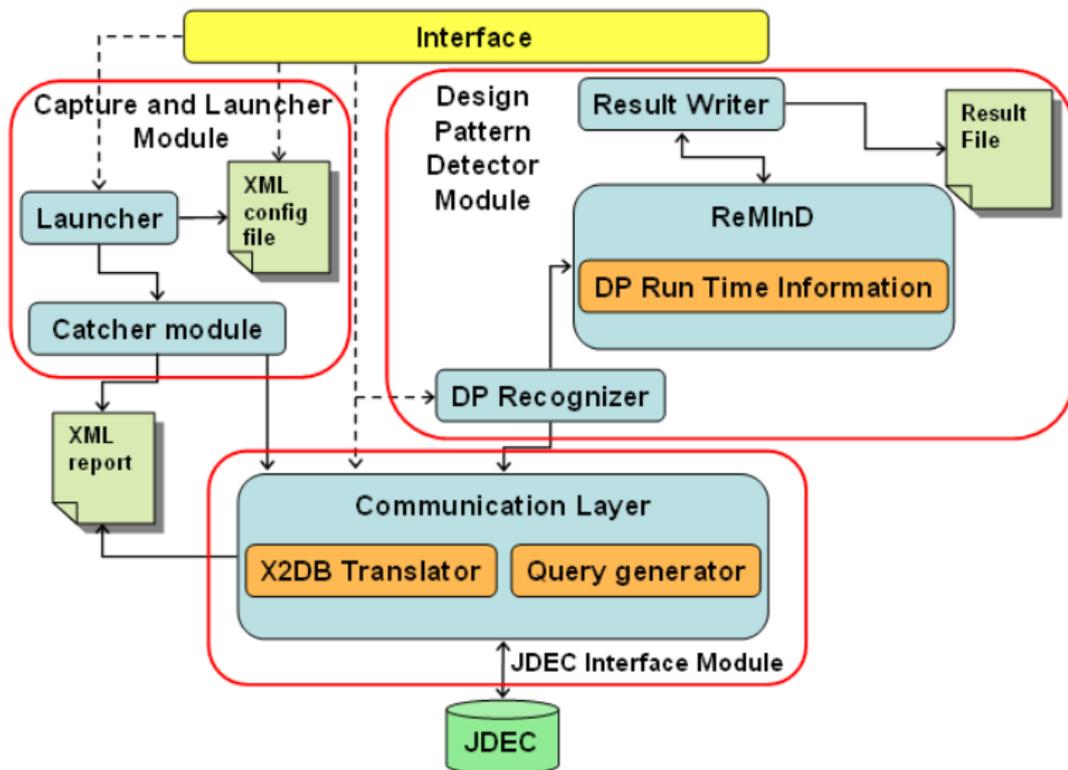
3 JADEPT architecture

4 Results

5 Conclusions

6 Future Development

JADEPT architecture



JDEC - Java Dynamic Event Collector

- JDEC is a database designed to store all information collected by the catcher module
- JDEC static and dynamic aspects of Java code are modelled in JDEC structure
- Database advantages:
 - A large number of data can be stored
 - Different executions can be stored
 - Some rules can be translated directly in queries

1 Objective and Specifications

2 Rules

3 JADEPT architecture

4 Results

5 Conclusions

6 Future Development

Results

Dynamic
analysis for
Design
Pattern
detection in
Java:
Information
collected using
JPDA

Fabrizio Perin

Outline

Objective and
Specifications

Rules

JADEPT
architecture

Results

Conclusions

Future
Development

COR impl.	COR Detector	Observer Detector	Visitor detector
Cooper	100%	10%	17%
Earthlink	76%	0%	0%
Fluffycat	7%	0%	0%
Kuchana	69%	0%	0%

Results

Dynamic
analysis for
Design
Pattern
detection in
Java:
Information
collected using
JPDA

Fabrizio Perin

Outline

Objective and
Specifications

Rules

JADEPT
architecture

Results

Conclusions

Future
Development

Observer impl.	COR Detector	Observer Detector	Visitor detector
Cooper	15%	100%	20%
Earthlink	15%	0%	37%
Earthlink2	15%	10%	17%
Fluffycat	0%	0%	0%
Kuchana	23%	90%	40%
Sun	23%	47%	40%

Results

Dynamic analysis for Design Pattern detection in Java: Information collected using JPDA

Fabrizio Perin

Outline

Objective and Specifications

Rules

JADEPT architecture

Results

Conclusions

Future Development

Visitor impl.	COR Detector	Observer Detector	Visitor detector
Composite	7%	0%	93%
Composite3	69%	20%	93%
Cooper	76%	10%	37%
Earthlink	15%	0%	37%
Earthlink2	15%	10%	17%
Fluffycat	0%	0%	0%
Kuchana	23%	0%	100%
Vis1	23%	26%	20%
Sun	23%	26%	100%
VisitorContact	0%	26%	100%

Results

Dynamic
analysis for
Design
Pattern
detection in
Java:
Information
collected using
JPDA

Fabrizio Perin

Outline

Objective and
Specifications

Rules

JADEPT
architecture

Results

Conclusions

Future
Development

System name	COR Detector	Observer Detector	Visitor detector
JADEPT	100%	90%	17%

1 Objective and Specifications

2 Rules

3 JADEPT architecture

4 Results

5 Conclusions

6 Future Development

Conclusions

- A design pattern detector has been designed meeting the specifications
- A functioning prototype (JADEPT) has been developed
- JADEPT has demonstrated the potential of the proposed architecture, even though further refinements are necessary

① Objective and Specifications

② Rules

③ JADEPT architecture

④ Results

⑤ Conclusions

⑥ Future Development

Future Development

- Refine the rules in term of score and definition
- Improve the design pattern detector module
- Improve the catcher module performance: JPDA causes a major drop in performances of the analyzed executable on complex codes
- Evaluate the changes on JPDA to improve the performance and granularity of information collected by the catcher module

references



E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison Wesley, Reading MA, USA, 1994.



J. W. Cooper, *The design pattern Java companion*, Addison-Wesley, October 1998.