PINOT

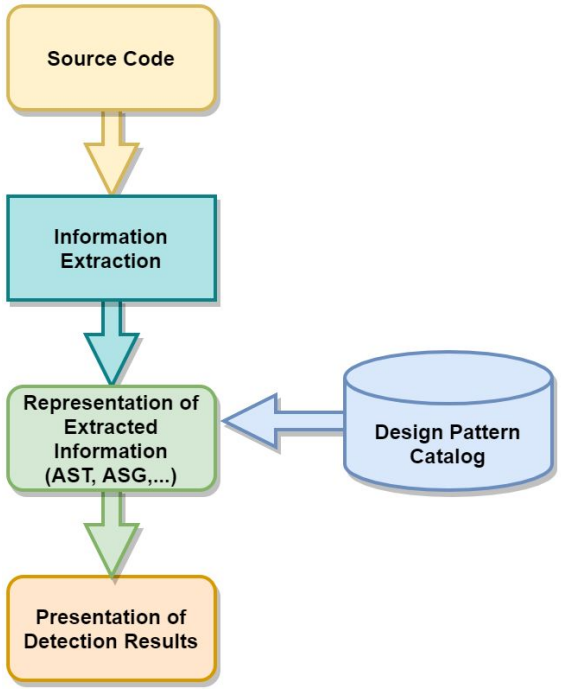
Pattern INference recOvery Tool

Swastik Satyanarayan Nayak & Sytse Oegema

Table of Contents

Pattern Recognition

- Structural aspects
- Behavioural aspects



pinot
Pattern INference and recOvery Tool

GoF Patterns

Creational Patterns
- Abstract Factory
- Builder
- Factory Method
- Prototype
- Singleton

Structural Patterns
- Adapter
- Bridge
- Composite
- Decorator
- Facade
- Flyweight
- Proxy

Behavioral Patterns
- Chain of Responsibility
- Command
- Interpreter
- Iterator
- Mediator
- Memento
- Observer
- State
- Strategy
- Template Method
- Visitor

pinot
Pattern INference and recOvery Tool

Design Patterns: Elements of Reusable Object-Oriented Software. Gamma et al. (1994)

Reclassification of GoF Patterns

Structure Driven
- Adapter
- Bridge
- Composite
- Facade
- Proxy
- Template Method
- Visitor

Behavior Driven
- Abstract Factory
- Chain of Responsibility
- Decorator
- Factory Method
- Flyweight
- Mediator
- Observer
- Singleton
- State
- Strategy

Language Provided
- Prototype
- Iterator

Domain Specific
- Command
- Interpreter

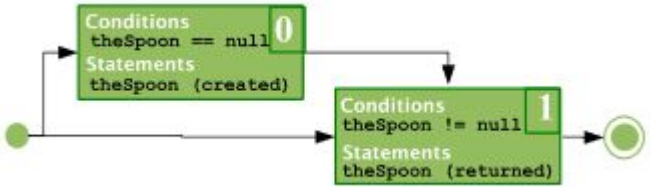
Generic Concepts
- Memento
- Builder

pinot
Pattern INference and recOvery Tool

Reverse Engineering of Design Patterns from Java Source Code. Nija Shi and Ronald A. Olsso(2006)

Methodology

- Data-Flow Analysis
- Control-Flow Graph(CFG)



pinot
Pattern INference and recOvery Tool

Reclassification of GoF Patterns

“interesting“

A tool for design pattern detection and software architecture reconstruction
Francesca Arcelli Fontana and Marco Zanoni (2011)

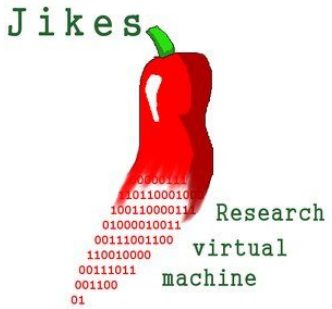
“interesting, but arguable“

Evaluation of Accuracy inDesign Pattern Occurrence Detection
Niklas Pettersson, Welf Löwe, and Joakim Nivre(2010)

pinot
Pattern INference and recOvery Tool

PINOT

- Java pattern recognition
- Based on Jikes - open source Java compiler
- Written in C++



pinot
Pattern INference and recOvery Tool

PINOT

```
1.  ------------------------------------
2.
3.  Pattern Instance Statistics:
4.
5.  Creational Patterns
6.  ============================
7.  Abstract Factory         7
8.  Factory Method           8
9.  Singleton                5
10. ----------------------------
11. Structural Patterns
12. ============================
13. Adapter                  3
14. Bridge                   2
15. Composite                5
16. Decorator                5
17. Facade                   13
18. Flyweight                1
19. Proxy                    10
20. ----------------------------
21. Behavioral Patterns
22. ============================
23. Chain of Responsibility  0
24. Mediator                 84
25. Observer                 12
26. State                    3
27. Strategy                 40
28. Template Method          1
29. Visitor                  0
30. ----------------------------
31.
32. Number of classes processed: 442
33. Number of files processed: 540
34. Size of DelegationTable: 2012
35. Size of concrete class nodes: 370
36. Size of undirected invocation edges: 213
```

pinot
Pattern INference and recOvery Tool

https://github.com/premaseem/designPatterns. premaseem/designPatterns(2017)

PINOT

1. Singleton Pattern
2. IvoryTower is a Singleton class
3. INSTANCE is the Singleton instance
4. getInstance creates and returns INSTANCE
5. File location: ../java/com/iluwatar/singleton/IvoryTower.java

https://github.com/premaseem/designPatterns. premaseem/designPatterns(2017)

Program Comprehension

Goal

- Analyze
- Refactor

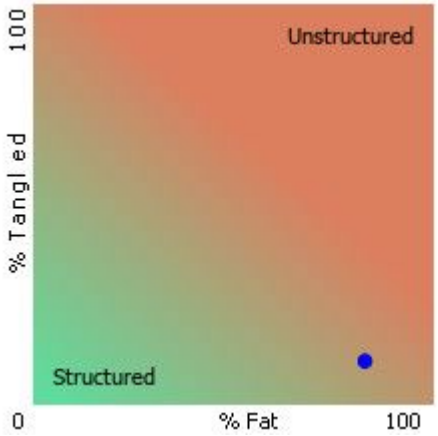
Bottom-Up Approach

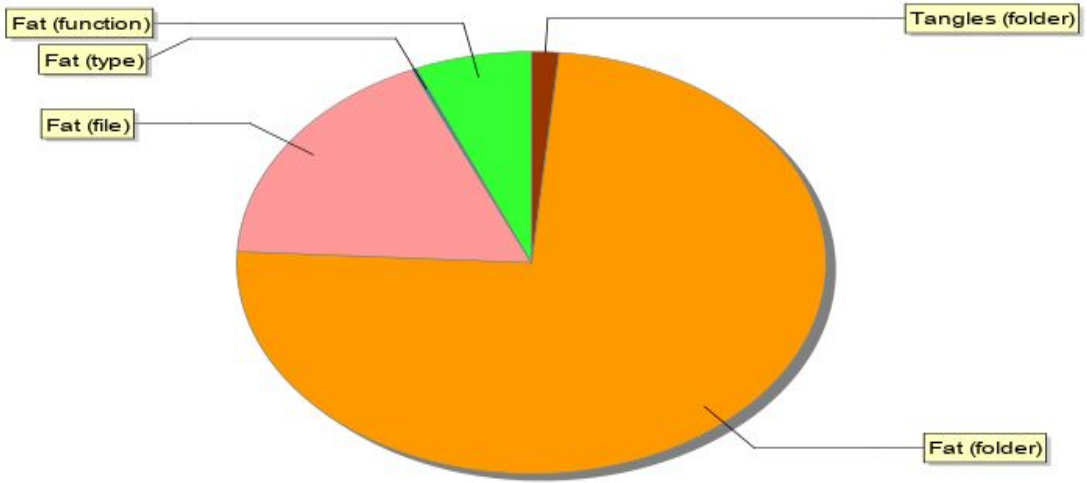
- Structure101 Generic & C++
- Understand

Files	69 (210)
Types	767
Functions	6956
Lines of Code(approx)	88 K

pinot
Pattern INference and recOvery Tool

Structure of PINOT





Technique: McCabe's metric, or [Cyclomatic Complexity]

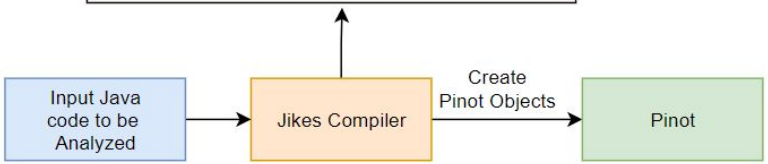
pinot
Pattern INference and recOvery Tool

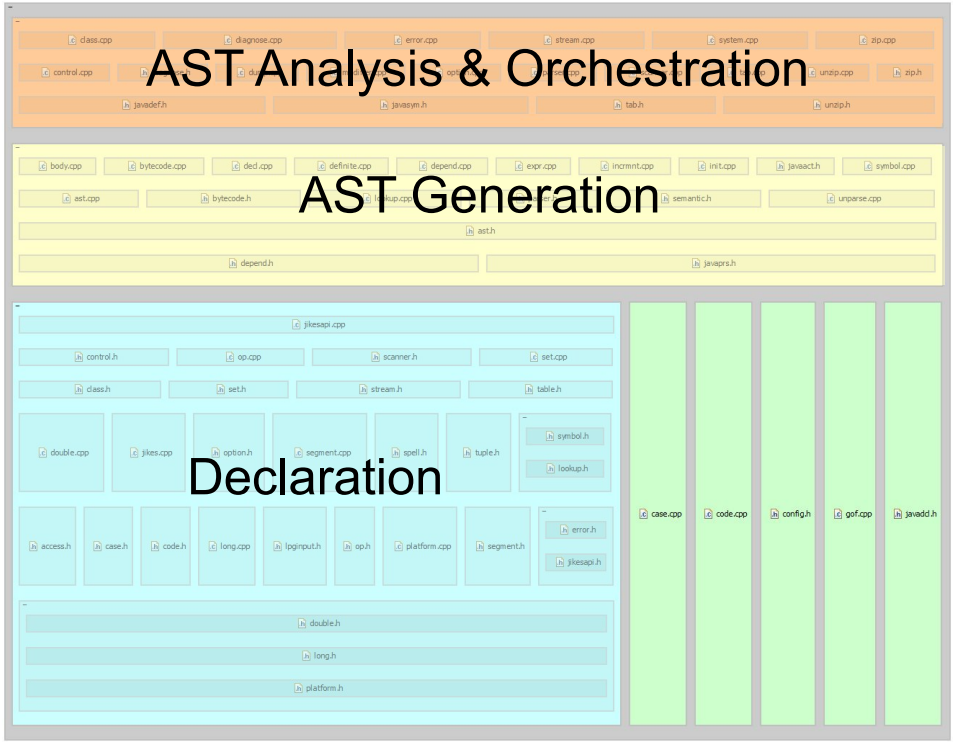
Structure of PINOT

The entry Point

RULES/Java Grammar

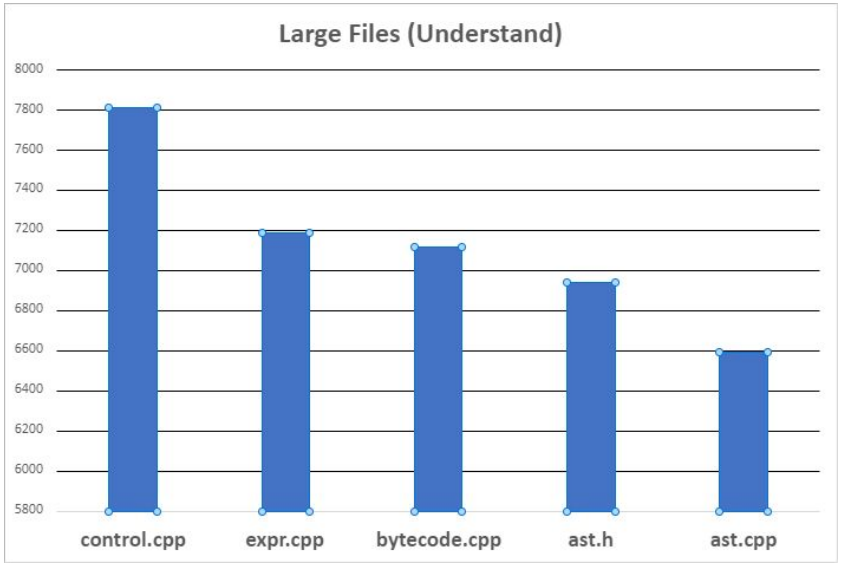
```
//
// Rule 3:  Literal ::= IntegerLiteral
//
#line 866 "java.g"
void Parser::Act3()
{
    Sym(1) = ast_pool -> NewIntegerLiteral(Token(1));
}
```

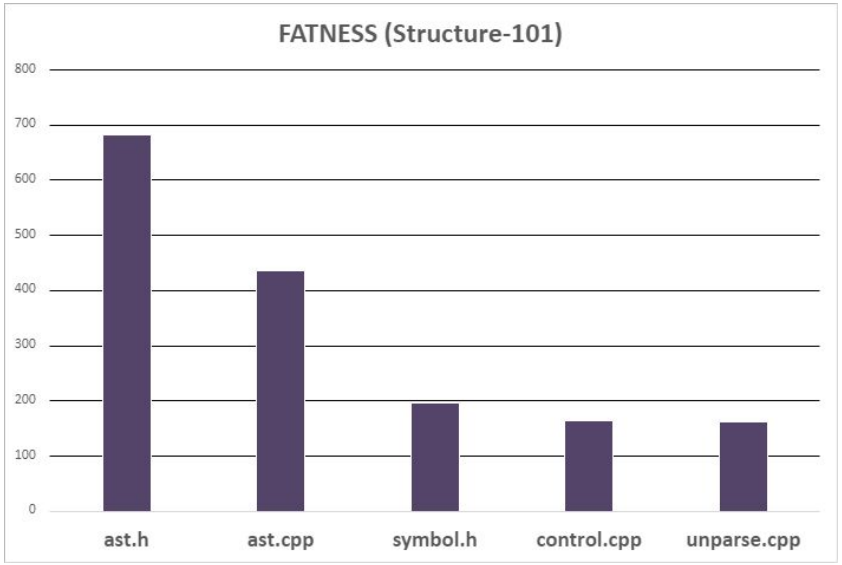




pinot
Pattern INference and recOvery Tool

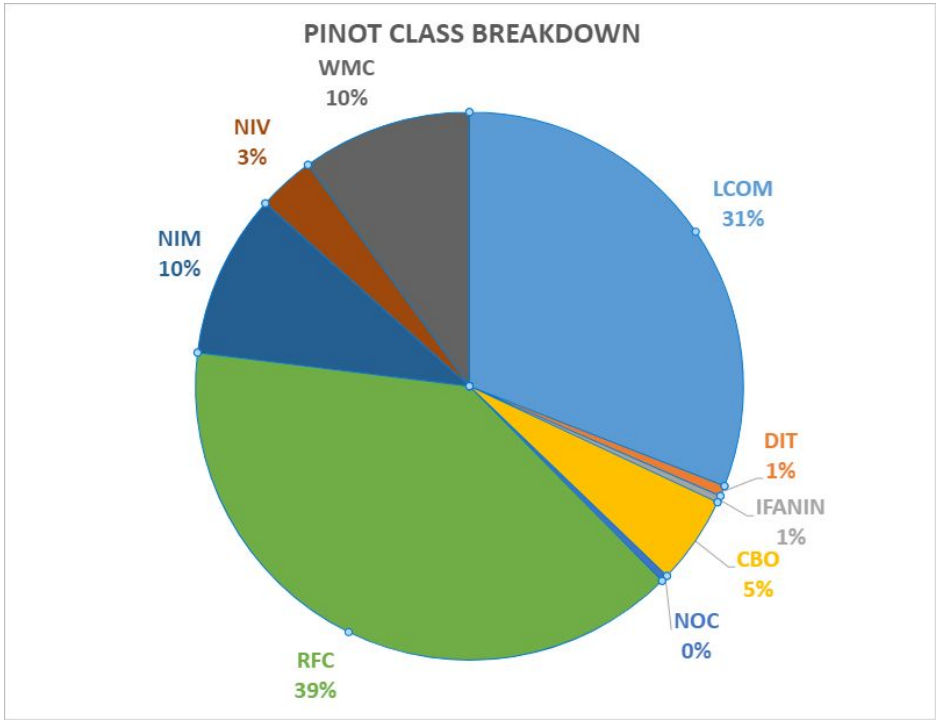
Analysis of Pinot's Fatness

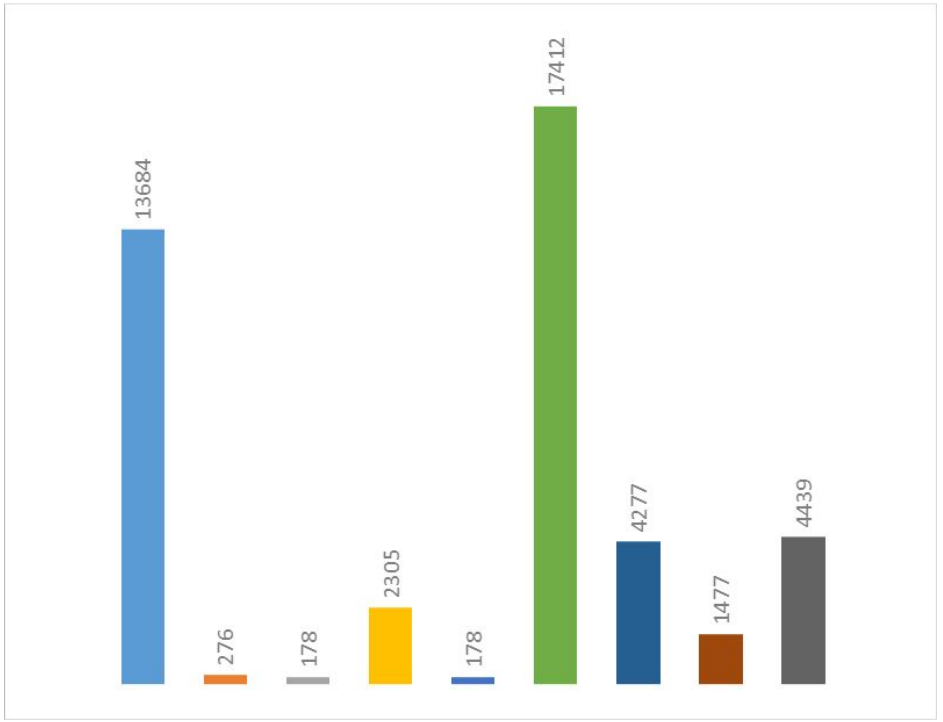




pinot
Pattern INference and recOvery Tool

Pinot Quality Attributes





48% of the Code lacks Cohesion (Median)

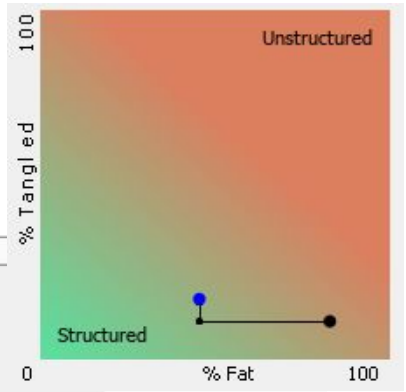
pinot
Pattern INference and recOvery Tool

LCOM (Percent Lack of Cohesion)	DIT (Max Inheritance Tree)	IFANIN (Count of Base Classes)	CBO (Count of Coupled Classes)	NOC (Count of Derived Classes)	RFC (Count of All Methods)	NIM (Count of Instance Methods)	NIV (Count of Instance Variables)	WMC (Count of Methods)

Refactor Plan

Strategy	Refactoring
Auto Levelize	Clustering source files in folders
Reduce fatness of files	
ast.h & ast.cpp	Move class StoragePool, and other related classes
Control.h & control.cpp	Move functionality from Control constructor, remove unused methods

Refactor Plan



java.g

src

orchestration

class.cpp diagnose.cpp error.cpp stream.cpp system.cpp zip.cpp

control.cpp diagnose.h dump.cpp modifier.cpp option.cpp parser.cpp scanner.cpp tab.cpp unzip.cpp

javadef.h javasym.h tab.h unzip.h

AST_generation

(20)

ast.cpp body.cpp bytecode.cpp decl.cpp definite.cpp depend.cpp expr.cpp incrmnt.cpp init.cpp javaact.h symbol.cpp unparse.cpp

bytecode.h lookup.cpp parser.h semantic.h storage

ast_parser

ast.h

depend.h javaprs.h

declarations

op.cpp

class.h scanner.h

stream.h tuple.h

jikes

control.h set.cpp

set.h table.h

double.cpp segment.cpp spell.h symbol

access.h case.h code.h long.cpp lpginput.h op.h platform.cpp segment.h

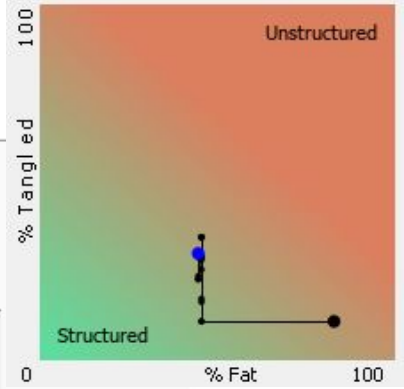
option.h platform

case.cpp code.cpp config.h

gof.cpp javadcl.h

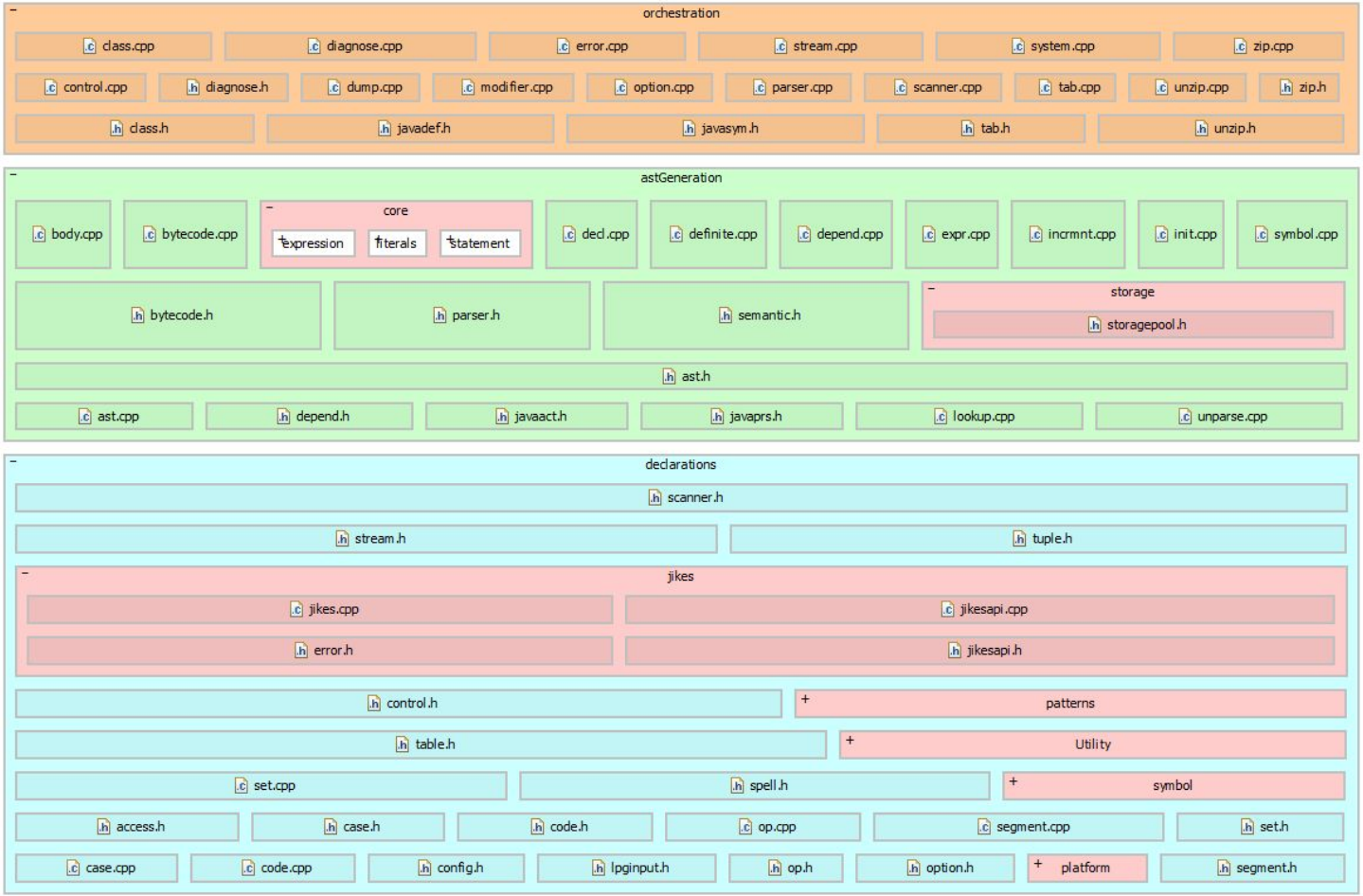
built-in mnt usr

Refactor Plan (Extended)



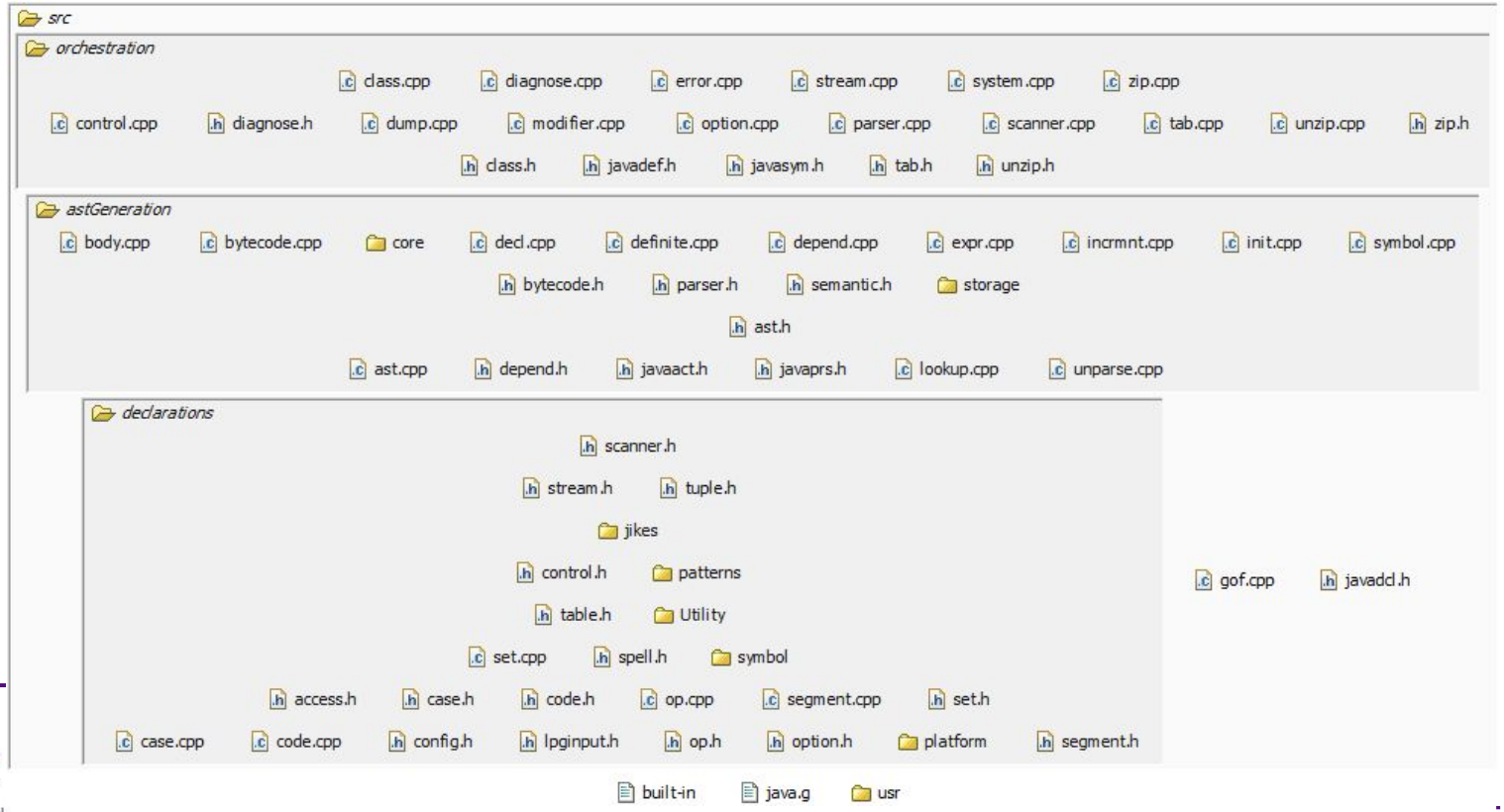
pinot
Pattern INference and recOvery Tool

Pinot Refactored Architecture



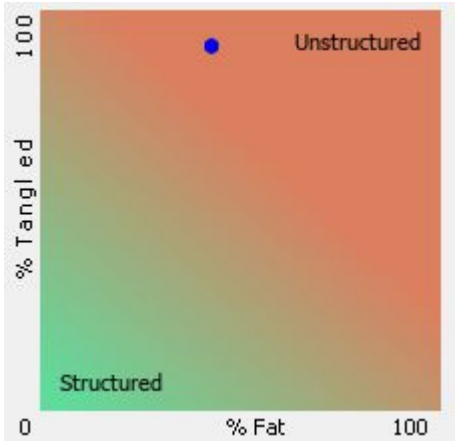
pinot
Pattern INference and recOvery Tool

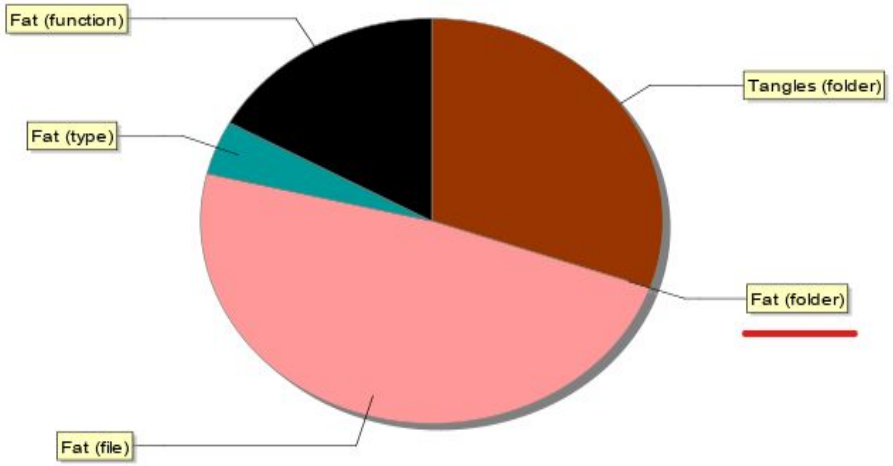
Refactored Pinot in Structure 101



pinot
Pattern INference and recOvery Tool

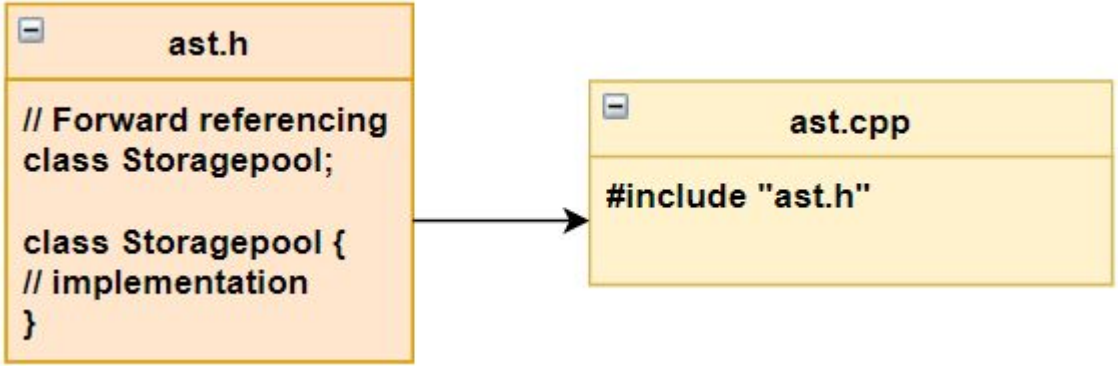
Structure of Pinot Refactored



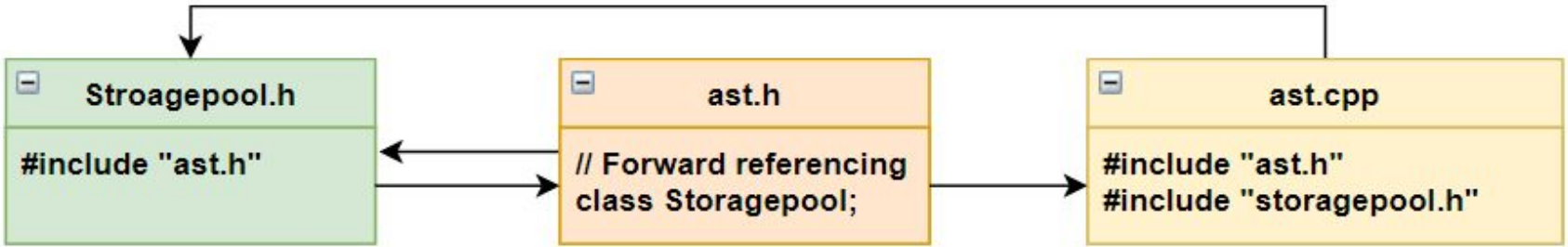


pinot
Pattern INference and recOvery Tool

Primary reason for Tangles



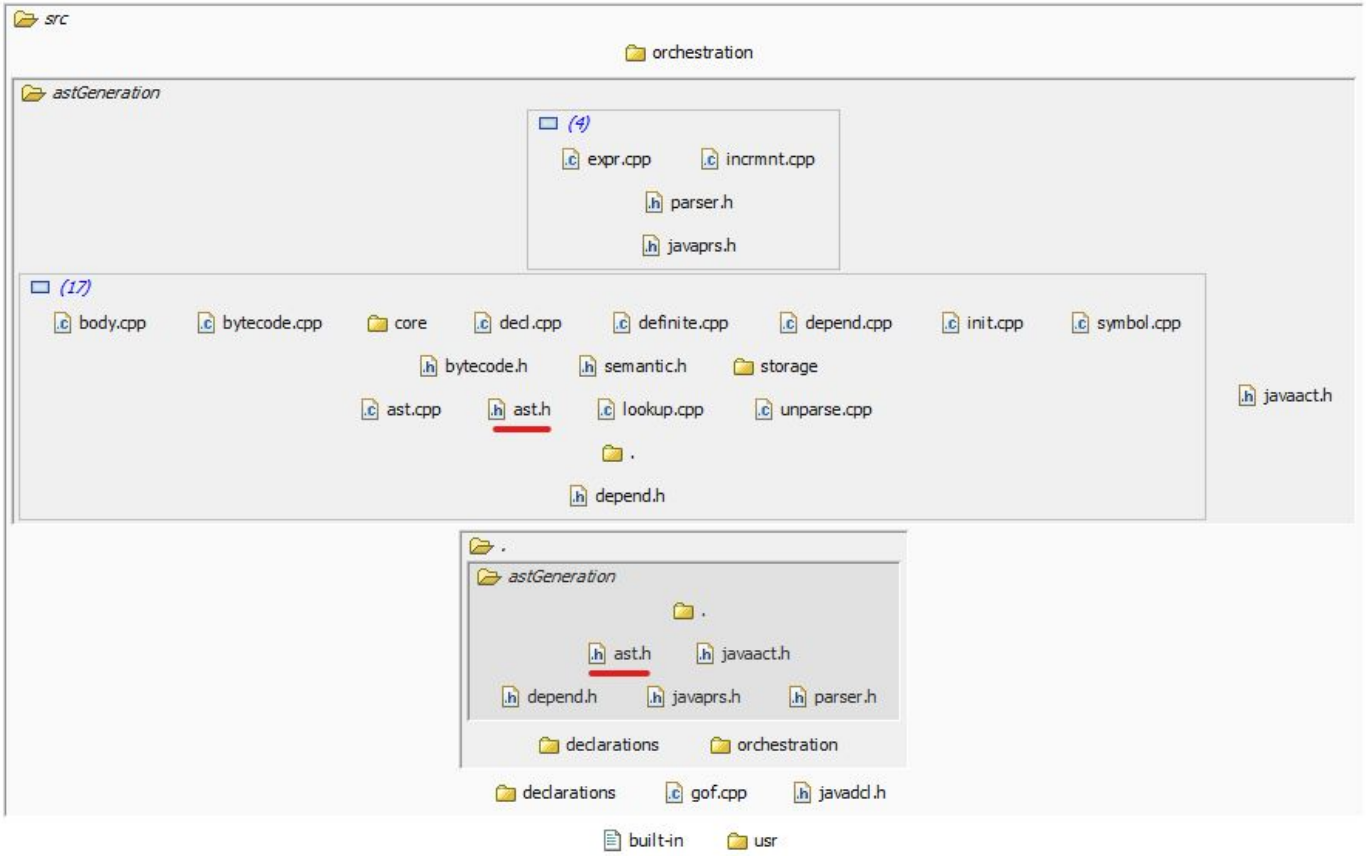
Before Refactoring



After Refactoring

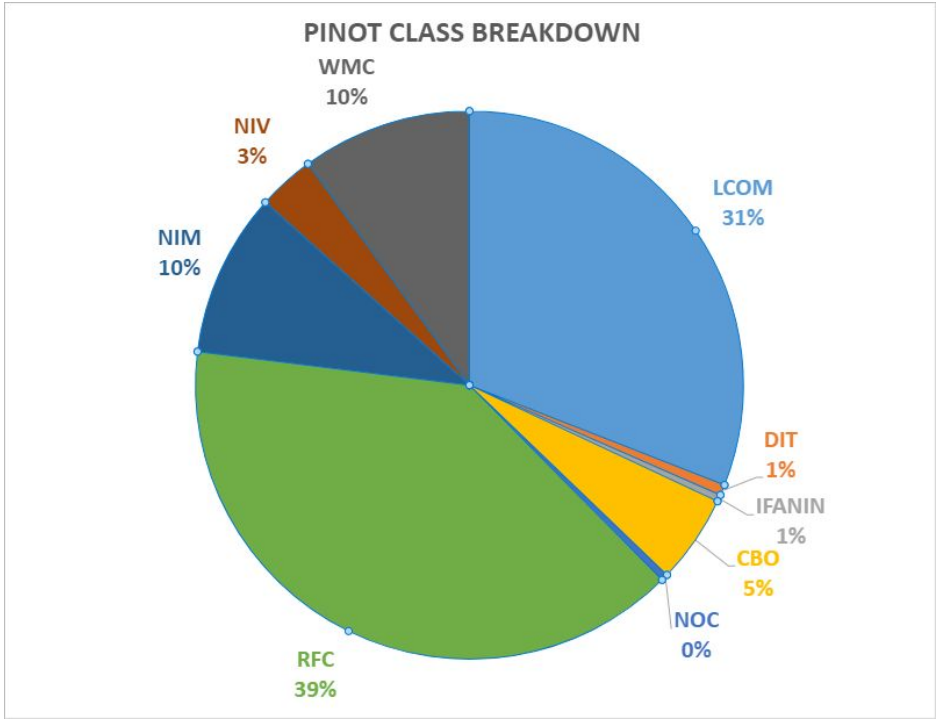
pinot
Pattern INference and recOvery Tool

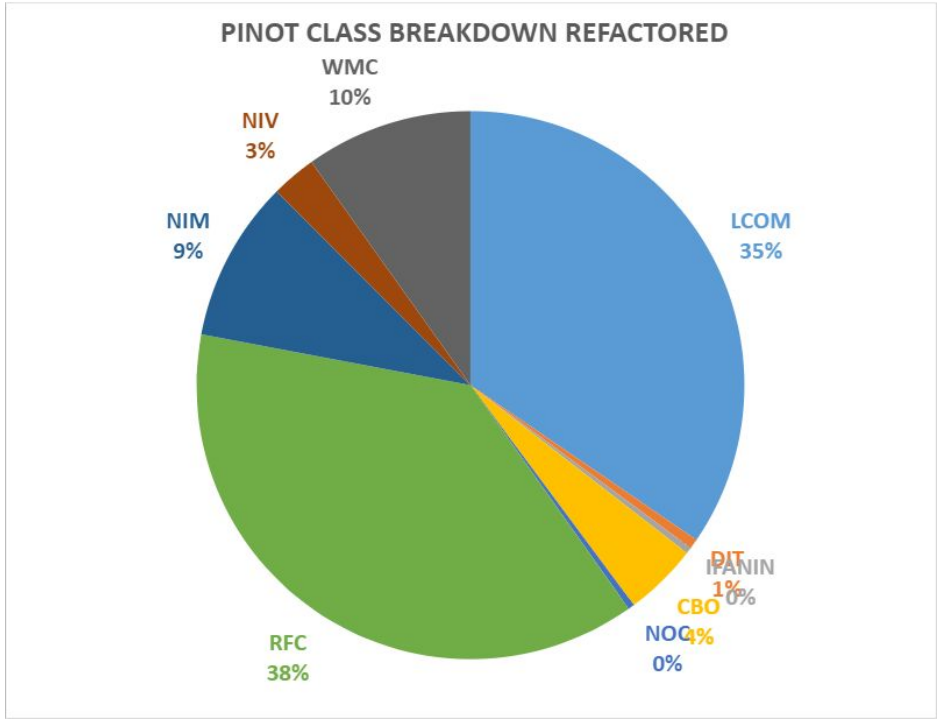
Another possible Culprit



pinot
Pattern INference and recOvery Tool

Quality Attributes of Pinot Refactored

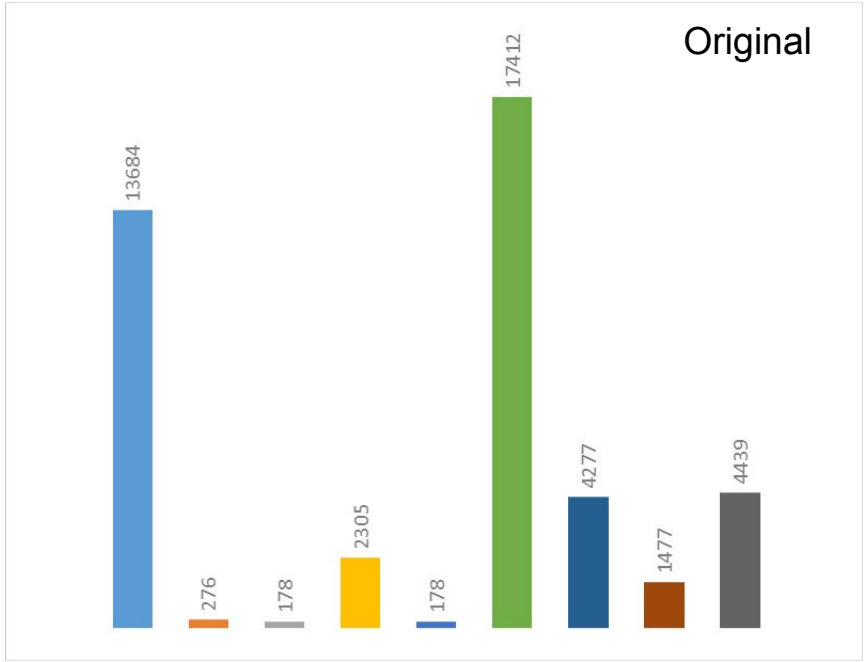




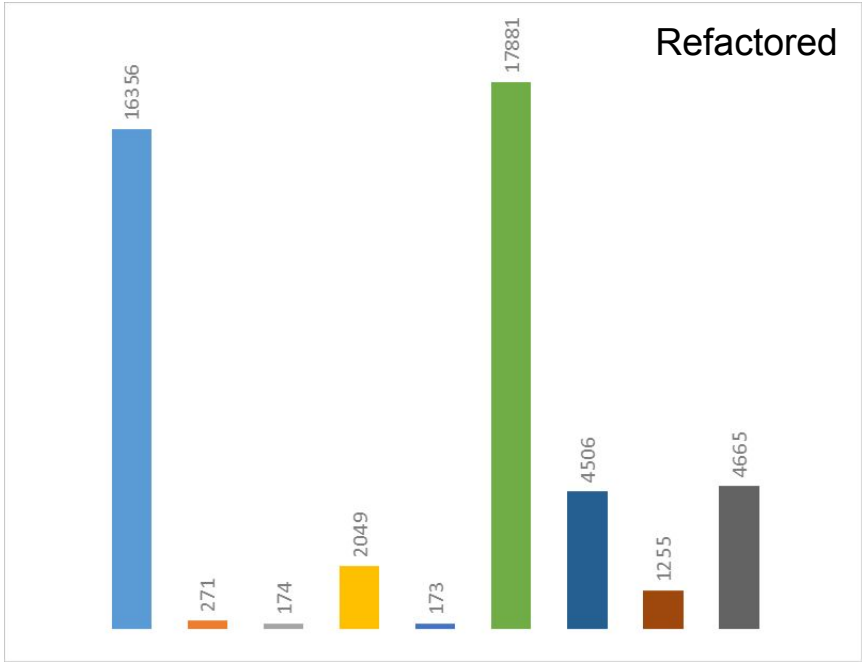
pinot
Pattern INference and recOvery Tool

LCOM (Percent Lack of Cohesion)	DIT (Max Inheritance Tree)	IFANIN (Count of Base Classes)	CBO (Count of Coupled Classes)	NOC (Count of Derived Classes)	RFC (Count of All Methods)	NIM (Count of Instance Methods)	NIV (Count of Instance Variables)	WMC (Count of Methods)

Quality Attributes of Pinot Refactored



48% of the Code lacks Cohesion (Median)



60% of the Code lacks Cohesion (Median)

pinot
Pattern INference and recOvery Tool

Evaluation

Original

```
Pattern Instance Statistics:

Creational Patterns
==============================
Abstract Factory              7
Factory Method                8
Singleton                     5
------------------------------
Structural Patterns
==============================
Adapter                       3
Bridge                        2
Composite                     5
Decorator                     5
Facade                       13
Flyweight                     1
Proxy                        10
------------------------------
Behavioral Patterns
==============================
Chain of Responsibility       0
Mediator                     84
Observer                     12
State                         3
Strategy                     40
Template Method               1
Visitor                       0
------------------------------

Number of classes processed: 442
Number of files processed: 540
Size of DelegationTable: 2012
Size of concrete class nodes: 370
Size of undirected invocation edges: 213


nMediatorFacadeDual/nMediator = 1/84
nImmutable/nFlyweight = 0/1
nFlyweightGoFVersion = 0
```

Refactored

```
------------------------------------------

Pattern Instance Statistics:

Creational Patterns
==============================
Abstract Factory              7
Factory Method                8
Singleton                     5
------------------------------
Structural Patterns
==============================
Adapter                       3
Bridge                        2
Composite                     5
Decorator                     5
Facade                       13
Flyweight                     1
Proxy                        10
------------------------------
Behavioral Patterns
==============================
Chain of Responsibility       0
Mediator                     84
Observer                     12
State                         3
Strategy                     40
Template Method               1
Visitor                       0
------------------------------

Number of classes processed: 442
Number of files processed: 540
Size of DelegationTable: 2012
Size of concrete class nodes: 370
Size of undirected invocation edges: 213


nMediatorFacadeDual/nMediator = 1/84
nImmutable/nFlyweight = 0/1
nFlyweightGoFVersion = 0
```

pinot
Pattern INference and recOvery Tool

Challenges

- Cyclic dependencies
- Excessive use of Class forwarding
- Higher degree of closely coupled logic in header files
- Custom memory allocation decision leading to Segmentation errors
- Some incompatibility with modern systems and/or compilers
- Support for Jikes has been discontinued as of 2010

Pinot vs Modern Java code

The code base [2] contains:

- Singleton
- Controller
- Factory
- Strategy
- Proxy
- Observer
- Aspect-oriented programming

Jikes[1]

Developer(s)	IBM
Stable release	**1.22** / October 3, 2004; 16 years ago
Operating system	Cross-platform
Type	Java compiler **Java 5.0**
License	IBM Public License
Website	jikes.sourceforge.net

```
-------------------------------------------

Pattern Instance Statistics:

Creational Patterns
============================
Abstract Factory            0
Factory Method              0
Singleton                   0
----------------------------
Structural Patterns
============================
Adapter                     0
Bridge                      0
Composite                   0
Decorator                   0
Facade                      0
Flyweight                   0
Proxy                       0
----------------------------
Behavioral Patterns
============================
Chain of Responsibility     0
Mediator                    0
Observer                    0
State                       0
Strategy                    0
Template Method             0
Visitor                     0
----------------------------

Number of classes processed: 1
Number of files processed: 15
Size of DelegationTable: 0
Size of concrete class nodes: 0
Size of undirected invocation edges: 0


nMediatorFacadeDual/nMediator = 0/0
nImmutable/nFlyweight = 0/0
nFlyweightGoFVersion = 0
```

pinot
Pattern INference and recOvery Tool

[1] https://en.wikipedia.org/wiki/Jikes
[2] https://github.com/indrekru/design-patterns-spring-boot

References

1. Pinot: https://www.cs.ucdavis.edu/~shini/research/pinot/
2. Reverse Engineering of Design Patterns from Java Source Code https://www.cs.ucdavis.edu/~shini/research/pinot/reverseJavaPatterns.pdf
3. Reverse Engineering of Design Patterns for High Performance Computing https://www.cs.ucdavis.edu/~olsson/pubs/2005/shi.pdf
4. Reverse Engineering of Design Patterns from Java Source Code https://www.cs.ucdavis.edu/~shini/research/pinot/pinot-ase06.ppt
5. Reverse Engineering of Design Patterns from Java Source Code https://www.cs.ucdavis.edu/~shini/research/pinot/pinot.ppt
6. Reverse Engineering of Design Patterns for High Performance Computing http://charm.cs.uiuc.edu/patHPC/slides/shi.pdf