DCC / ICEx / UFMG

Tools for Bad Smell Detection

Eduardo Figueiredo

http://www.dcc.ufmg.br/~figueiredo

Tools

- JDeodorant
- inFusion
- Stench Blossom
- PMD

JDeodorant

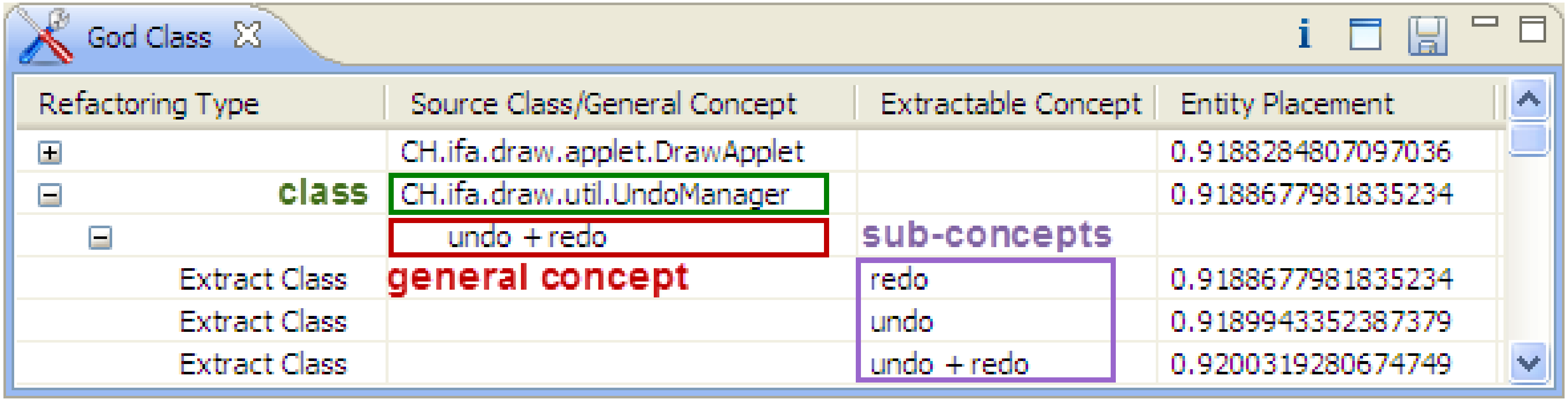
https://github.com/tsantalis/JDeodorant

JDeodorant

- Highlights
  - Open source project
  - Analyzes Java code
  - Eclipse plugin
- Bad smells
  - Feature Envy
  - Long Method
  - God Class

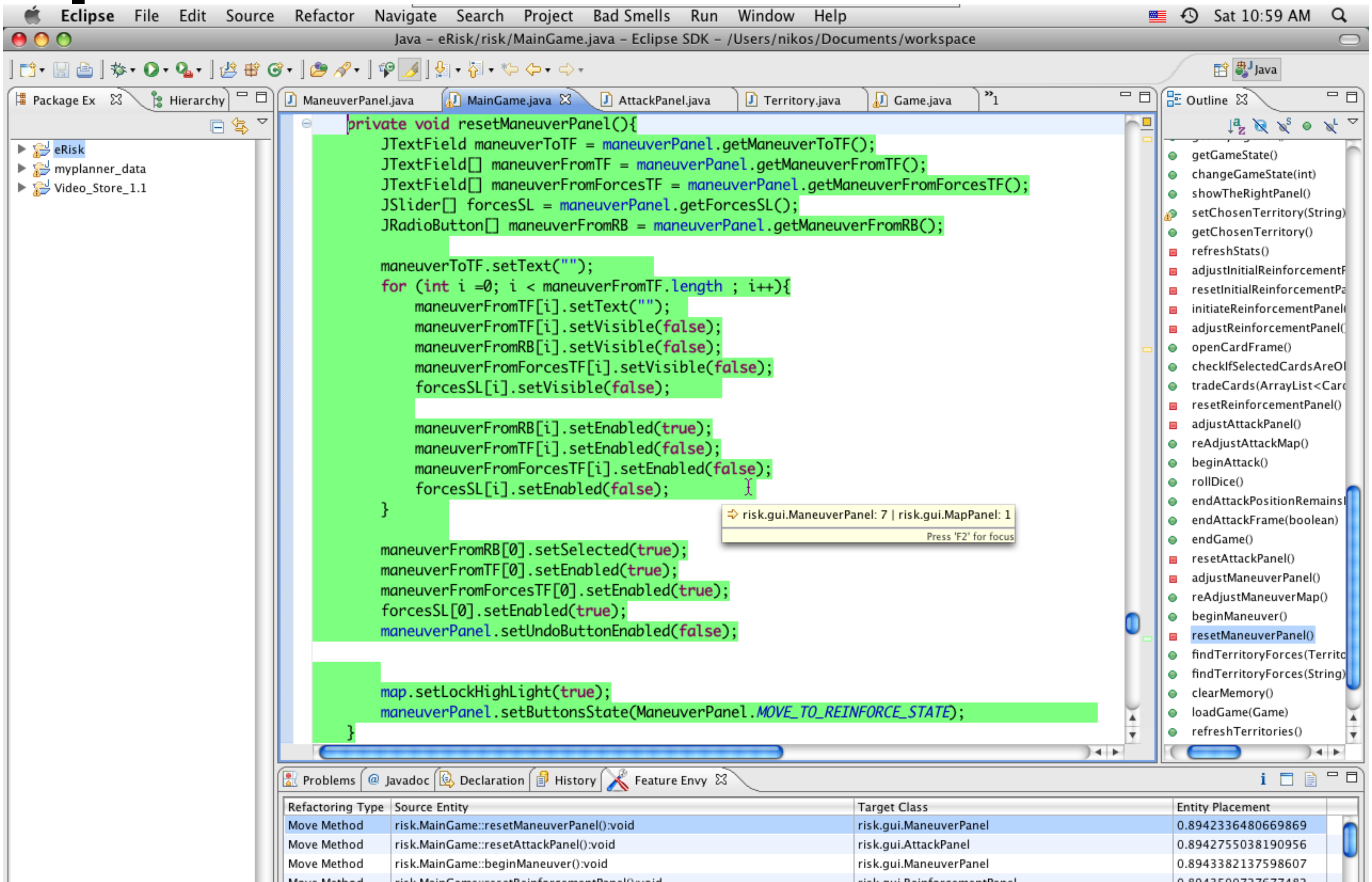
Screenshot: God Class

- Refactorings are recommended for each detected bad smell instance



Refactoring Type	Source Class/General Concept	Extractable Concept	Entity Placement
⊞	CH.ifa.draw.applet.DrawApplet		0.9188284807097036
⊟	CH.ifa.draw.util.UndoManager		0.9188677981835234
⊟	undo + redo		
Extract Class		redo	0.9188677981835234
Extract Class		undo	0.9189943352387379
Extract Class		undo + redo	0.9200319280674749

Screenshot: Feature Envy



```
private void resetManeuverPanel(){
    JTextField maneuverToTF = maneuverPanel.getManeuverToTF();
    JTextField[] maneuverFromTF = maneuverPanel.getManeuverFromTF();
    JTextField[] maneuverFromForcesTF = maneuverPanel.getManeuverFromForcesTF();
    JSlider[] forcesSL = maneuverPanel.getForcesSL();
    JRadioButton[] maneuverFromRB = maneuverPanel.getManeuverFromRB();

    maneuverToTF.setText("");
    for (int i =0; i < maneuverFromTF.length ; i++){
        maneuverFromTF[i].setText("");
        maneuverFromTF[i].setVisible(false);
        maneuverFromRB[i].setVisible(false);
        maneuverFromForcesTF[i].setVisible(false);
        forcesSL[i].setVisible(false);

        maneuverFromRB[i].setEnabled(true);
        maneuverFromTF[i].setEnabled(false);
        maneuverFromForcesTF[i].setEnabled(false);
        forcesSL[i].setEnabled(false);
    }

    maneuverFromRB[0].setSelected(true);
    maneuverFromTF[0].setEnabled(true);
    maneuverFromForcesTF[0].setEnabled(true);
    forcesSL[0].setEnabled(true);
    maneuverPanel.setUndoButtonEnabled(false);


    map.setLockHighLight(true);
    maneuverPanel.setButtonsState(ManeuverPanel.MOVE_TO_REINFORCE_STATE);
}
```

⇨ risk.gui.ManeuverPanel: 7 | risk.gui.MapPanel: 1

Press 'F2' for focus

Refactoring Type	Source Entity	Target Class	Entity Placement
Move Method	risk.MainGame::resetManeuverPanel():void	risk.gui.ManeuverPanel	0.8942336480669869
Move Method	risk.MainGame::resetAttackPanel():void	risk.gui.AttackPanel	0.8942755038190956
Move Method	risk.MainGame::beginManeuver():void	risk.gui.ManeuverPanel	0.8943382137598607
Move Method	risk.MainGame::resetReinforcementPanel():void	risk.gui.ReinforcementPanel	0.8943500727677483

inFusion

inFusion

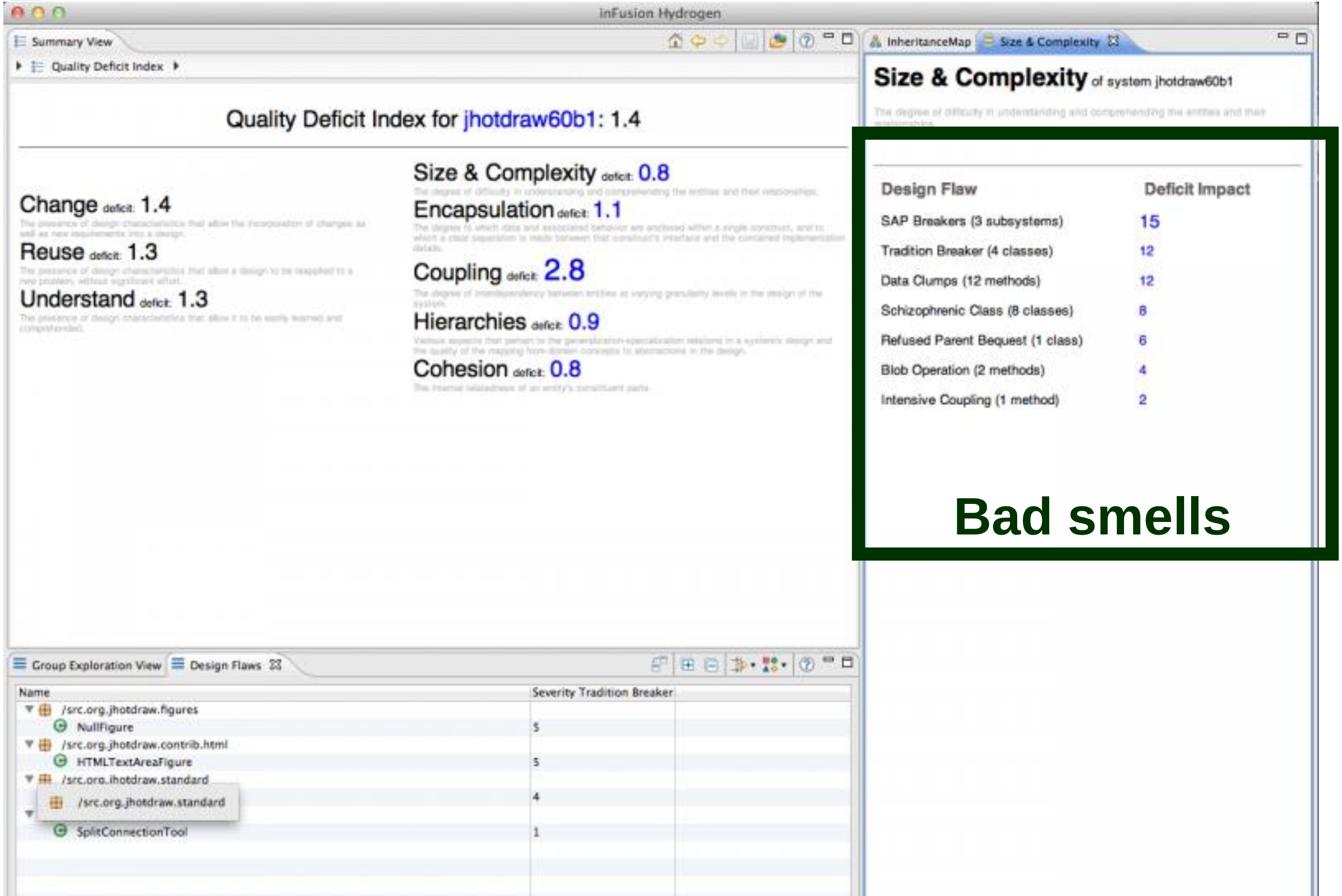
http://www.intooitus.com/products/infusion

https://marketplace.eclipse.org/content/infusion-hydrogen

inFusion

- Highlights
  - It supports C, C++ and Java
  - It includes visualization (polymetric views)
- Bad smells
  - Duplicated code
  - Feature Envy
  - God Class, etc.

Screenshot: Bad Smells



Quality Deficit Index for jhotdraw60b1: 1.4

Change deficit: 1.4
The presence of design characteristics that allow the incorporation of changes as well as new requirements into a design.

Reuse deficit: 1.3
The presence of design characteristics that allow a design to be reapplied to a new problem, without significant effort.

Understand deficit: 1.3
The presence of design characteristics that allow it to be easily learned and comprehended.

Size & Complexity deficit: 0.8
The degree of difficulty in understanding and comprehending the entities and their relationships.

Encapsulation deficit: 1.1
The degree to which data and associated behavior are enclosed within a single construct, and to which a clear separation is made between that construct's interface and the contained implementation details.

Coupling deficit: 2.8
The degree of interdependency between entities at varying granularity levels in the design of the system.

Hierarchies deficit: 0.9
Various aspects that pertain to the generalization-specialization relations in a system's design and the quality of the mapping from domain concepts to abstractions in the design.

Cohesion deficit: 0.8
The internal relatedness of an entity's constituent parts.

Size & Complexity of system jhotdraw60b1

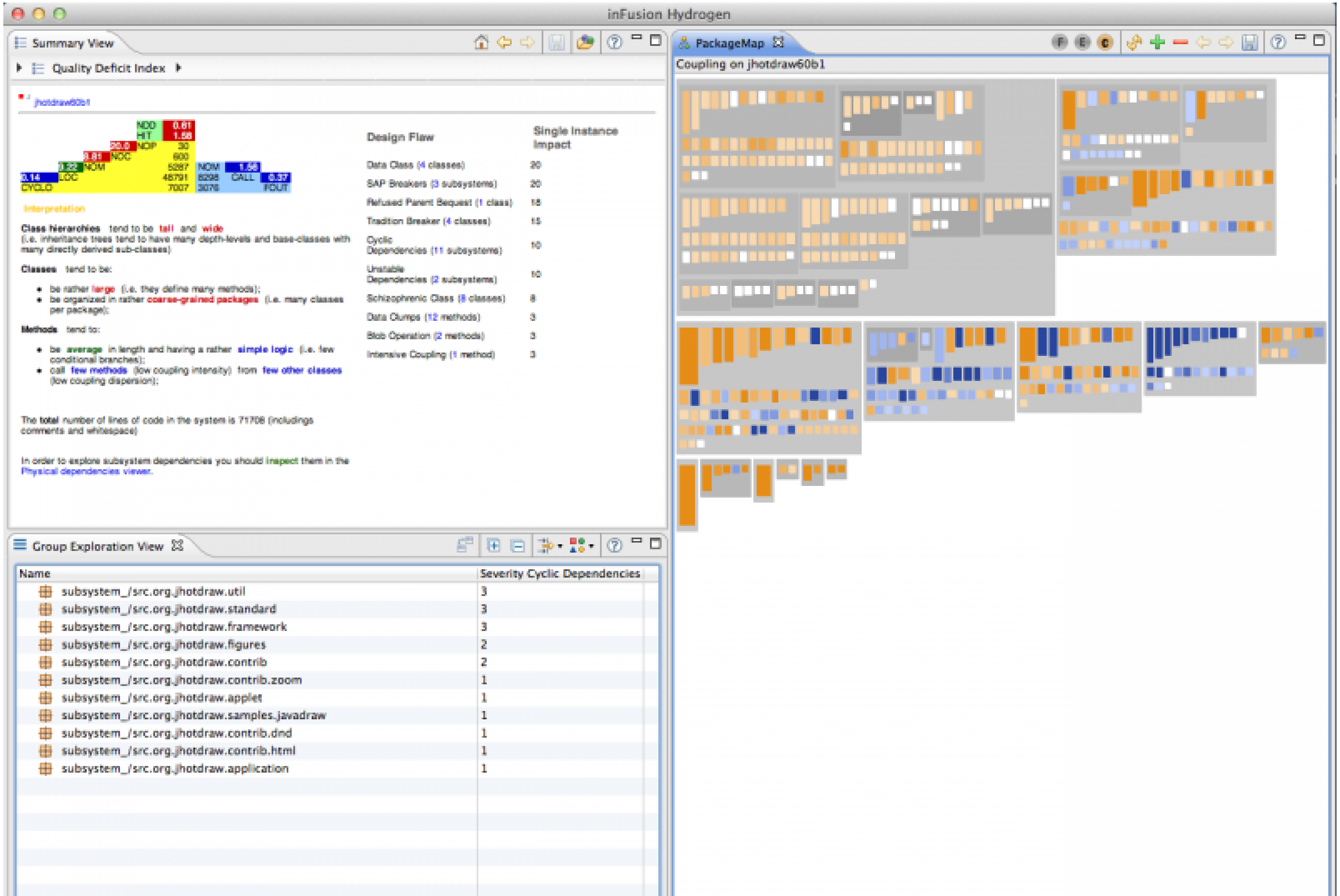
The degree of difficulty in understanding and comprehending the entities and their relationships.

Design Flaw	Deficit Impact
SAP Breakers (3 subsystems)	15
Tradition Breaker (4 classes)	12
Data Clumps (12 methods)	12
Schizophrenic Class (8 classes)	8
Refused Parent Bequest (1 class)	6
Blob Operation (2 methods)	4
Intensive Coupling (1 method)	2

Bad smells

Name	Severity Tradition Breaker
/src.org.jhotdraw.figures	
NullFigure	5
/src.org.jhotdraw.contrib.html	
HTMLTextAreaFigure	5
/src.org.jhotdraw.standard	
/src.org.jhotdraw.standard	4
SplitConnectionTool	1

Screenshot: Visualizations



Interpretation

Class hierarchies tend to be tall and wide (i.e. inheritance trees tend to have many depth-levels and base-classes with many directly derived sub-classes)

Classes tend to be:

- be rather large (i.e. they define many methods);
- be organized in rather coarse-grained packages (i.e. many classes per package);

Methods tend to:

- be average in length and having a rather simple logic (i.e. few conditional branches);
- call few methods (low coupling intensity) from few other classes (low coupling dispersion);

The **total** number of lines of code in the system is 71708 (includings comments and whitespace)

In order to explore subsystem dependencies you should inspect them in the Physical dependencies viewer.

Design Flaw	Single Instance Impact
Data Class (4 classes)	20
SAP Breakers (3 subsystems)	20
Refused Parent Bequest (1 class)	18
Tradition Breaker (4 classes)	15
Cyclic Dependencies (11 subsystems)	10
Unstable Dependencies (2 subsystems)	10
Schizophrenic Class (8 classes)	8
Data Clumps (12 methods)	3
Blob Operation (2 methods)	3
Intensive Coupling (1 method)	3

Name	Severity Cyclic Dependencies
subsystem_/src.org.jhotdraw.util	3
subsystem_/src.org.jhotdraw.standard	3
subsystem_/src.org.jhotdraw.framework	3
subsystem_/src.org.jhotdraw.figures	2
subsystem_/src.org.jhotdraw.contrib	2
subsystem_/src.org.jhotdraw.contrib.zoom	1
subsystem_/src.org.jhotdraw.applet	1
subsystem_/src.org.jhotdraw.samples.javadraw	1
subsystem_/src.org.jhotdraw.contrib.dnd	1
subsystem_/src.org.jhotdraw.contrib.html	1
subsystem_/src.org.jhotdraw.application	1

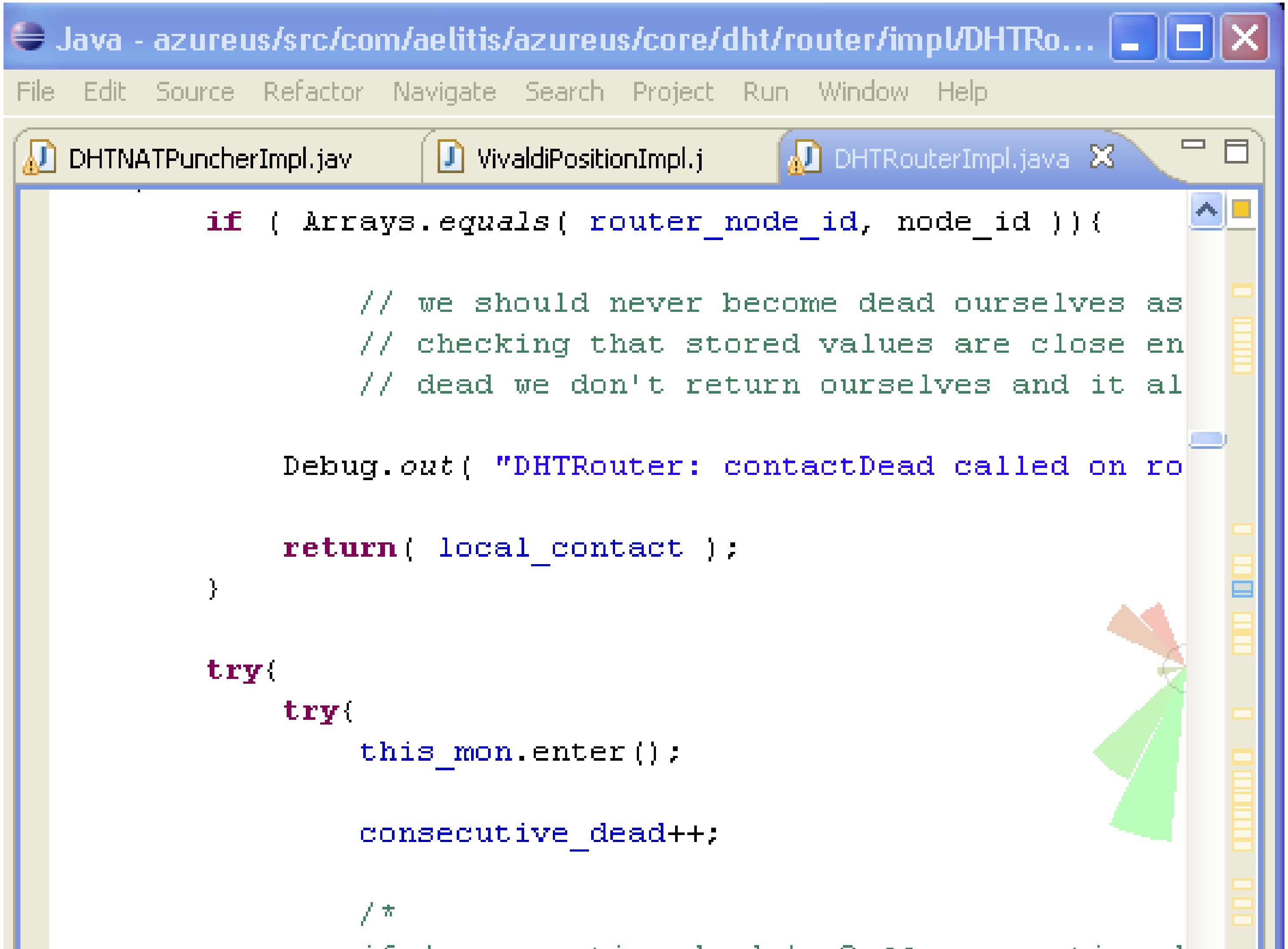
Stench Blossom

http://multiview.cs.pdx.edu/refactoring/smells/

Stench Blossom

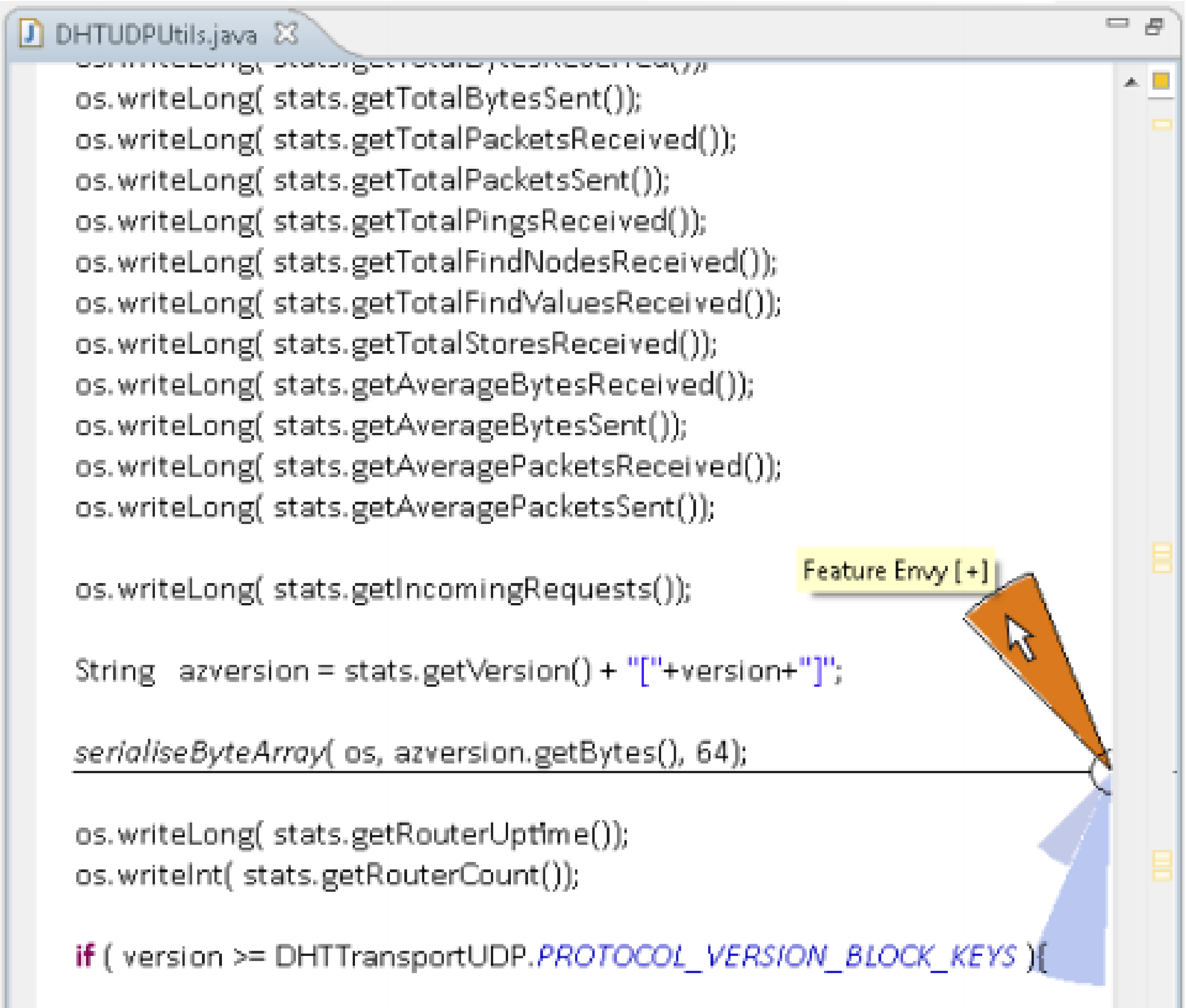
- Highlights
  - Open source project
  - Analyzes Java code
  - Eclipse plugin
  - Detect smells while coding
- Bad smells
  - Feature Envy
  - Data Clumps
  - Long Method, etc.

Screenshot: Editor



```
        if ( Arrays.equals( router_node_id, node_id )){

                // we should never become dead ourselves as
                // checking that stored values are close en
                // dead we don't return ourselves and it al

            Debug.out( "DHTRouter: contactDead called on ro

            return( local_contact );
        }

        try{
            try{
                this_mon.enter();

                consecutive_dead++;

                /*
```

Screenshot: Feature Envy



```
os.writeLong( stats.getTotalBytesReceived());
os.writeLong( stats.getTotalBytesSent());
os.writeLong( stats.getTotalPacketsReceived());
os.writeLong( stats.getTotalPacketsSent());
os.writeLong( stats.getTotalPingsReceived());
os.writeLong( stats.getTotalFindNodesReceived());
os.writeLong( stats.getTotalFindValuesReceived());
os.writeLong( stats.getTotalStoresReceived());
os.writeLong( stats.getAverageBytesReceived());
os.writeLong( stats.getAverageBytesSent());
os.writeLong( stats.getAveragePacketsReceived());
os.writeLong( stats.getAveragePacketsSent());

os.writeLong( stats.getIncomingRequests());

String  azversion = stats.getVersion() + "["+version+"]";

serialiseByteArray( os, azversion.getBytes(), 64);

os.writeLong( stats.getRouterUptime());
os.writeInt( stats.getRouterCount());

if ( version >= DHTTransportUDP.PROTOCOL_VERSION_BLOCK_KEYS ){
```

pmd

DON'T SHOOT THE MESSENGER

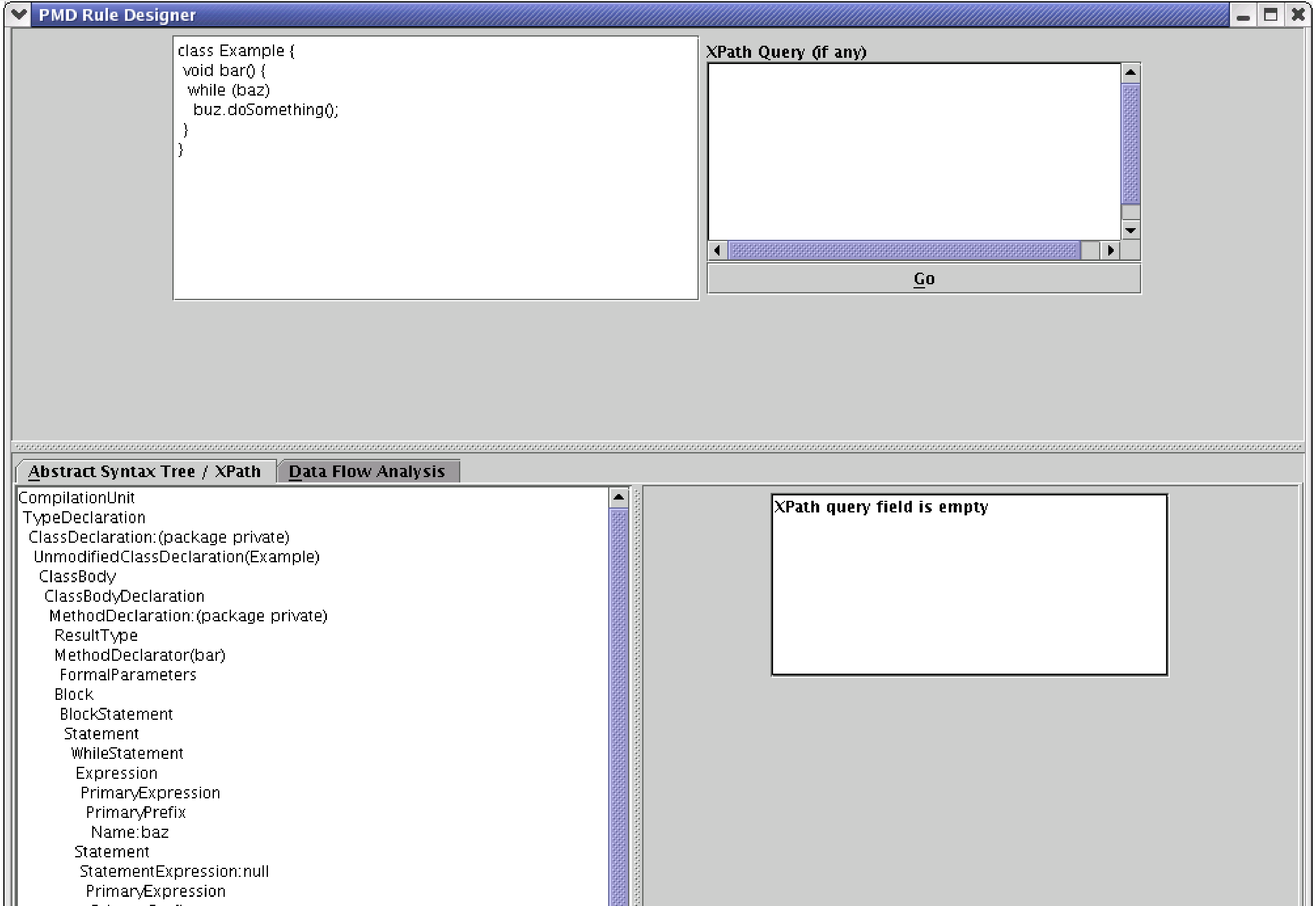
PMD

https://pmd.github.io/

PMD

- Highlights
  - Open source project
  - It supports Java, JavaScript, and others
  - PMD is integrated with IDE, like Eclipse
- Bad smells
  - God Class
  - Duplicated code
  - Long Parameter List, etc.

Screenshot: Rule Editor



```
class Example {
 void bar() {
  while (baz)
   buz.doSomething();
 }
}
```

```
CompilationUnit
 TypeDeclaration
  ClassDeclaration:(package private)
   UnmodifiedClassDeclaration(Example)
    ClassBody
     ClassBodyDeclaration
      MethodDeclaration:(package private)
       ResultType
       MethodDeclarator(bar)
        FormalParameters
       Block
        BlockStatement
         Statement
          WhileStatement
           Expression
            PrimaryExpression
             PrimaryPrefix
              Name:baz
           Statement
            StatementExpression:null
             PrimaryExpression
```

XPath query field is empty

Bibliography

- Websites of the Tools
- E. Murphy-Hill, A. P. Black. **An Interactive Ambient Visualization for Code Smells**. Proceedings of the 5th International Symposium on Software Visualization (SOFTVIS), pages 5-14, 2010.