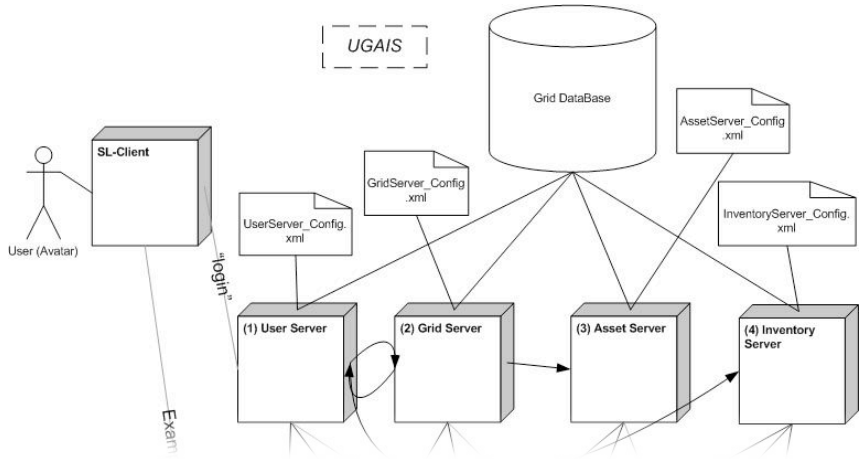
Understanding Large Codebases

```
public vo                ildDetail build, Map paramete
    Ite                  ).iterator();
    while (              {()) {
        try              parameters, (String) iterat
        }
    }
}

void assemblePlugin(BuildDetail build, Map parameters, String
    String className = line.trim();
    if (className.startsWith("#") || StringUtils.isEmpty(clas
        return;
    }
    Class clazz = Class.forName(className);
    Widget digesterService = (Widget) clazz.newInstance();
    mergeParameters(build, parameters);
    build.addPluginOutput(digesterService.getDisplayName(), c
            .getOutput(parameters));
}

private void mergeParameters(BuildDetail build, Map parameter
    parameters.put(Widget.PARAM_CC_ROOT, configuration.getCCH
```



micro <=> macro

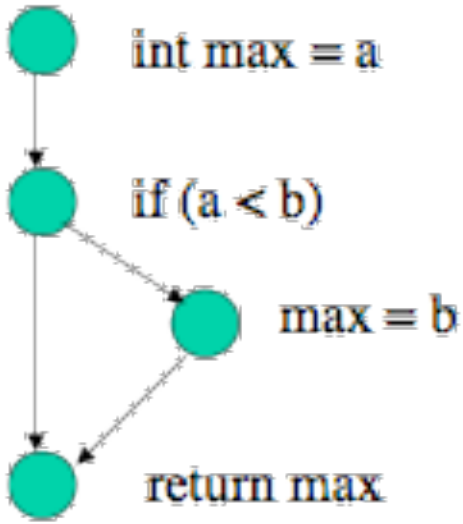

metrics

cyclomatic complexity

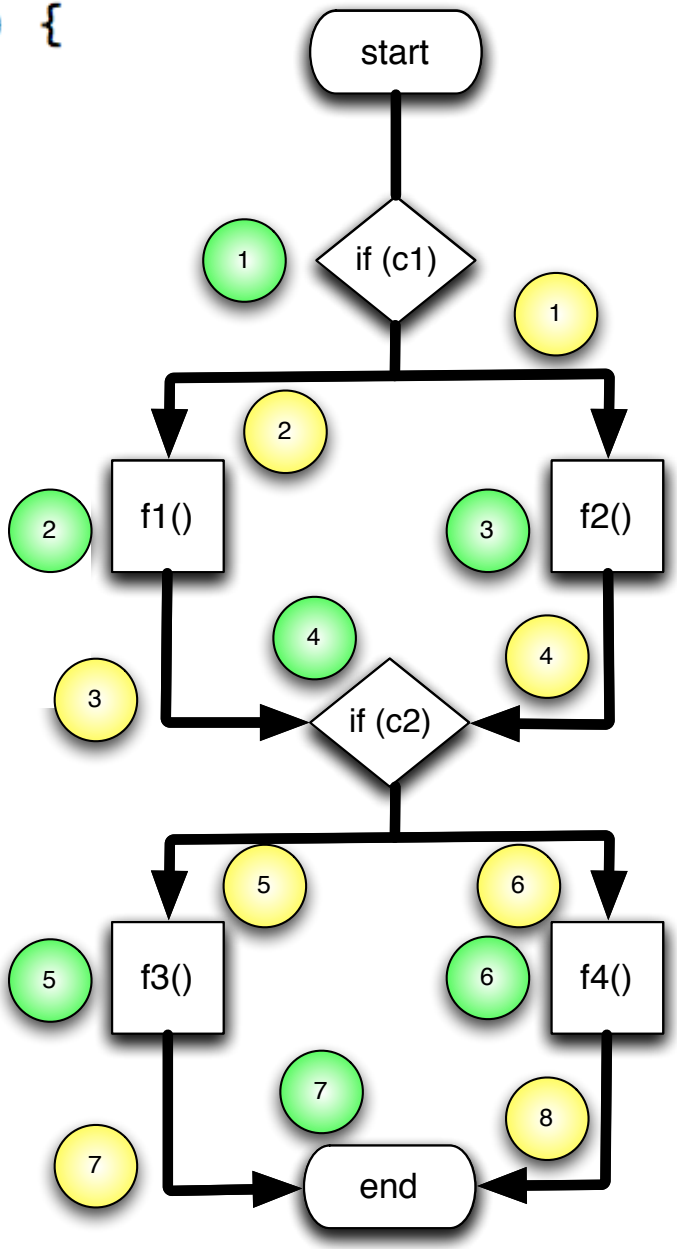
provides a numeric value representing the complexity of a function or method

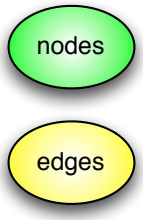
```
V(G)= e - n + 2
V(G) = cyclomatic complexity of G
e= # edges
n= # of nodes
```

```
int max (int a, int b) {
    int max = a;
    if (a < b) {
        max = b;
    }
    return max;
}
```



```
public void doIt() {
    if (c1) {
        f1();
    } else {
        f2();
    }
    if (c2) {
        f3();
    } else {
        f4();
    }
}
```





Chidamber & Kemerer object-oriented metrics

Shyam R. Chidamber
Chris F. Kemerer

http://faculty.salisbury.edu/~stlauterburg/cosc425/metricforood_chidamberkemerer94.pdf

easy (but trivial)

DIT	depth of inheritance tree
NOC	number of children
NPM	number of public methods

quite useful

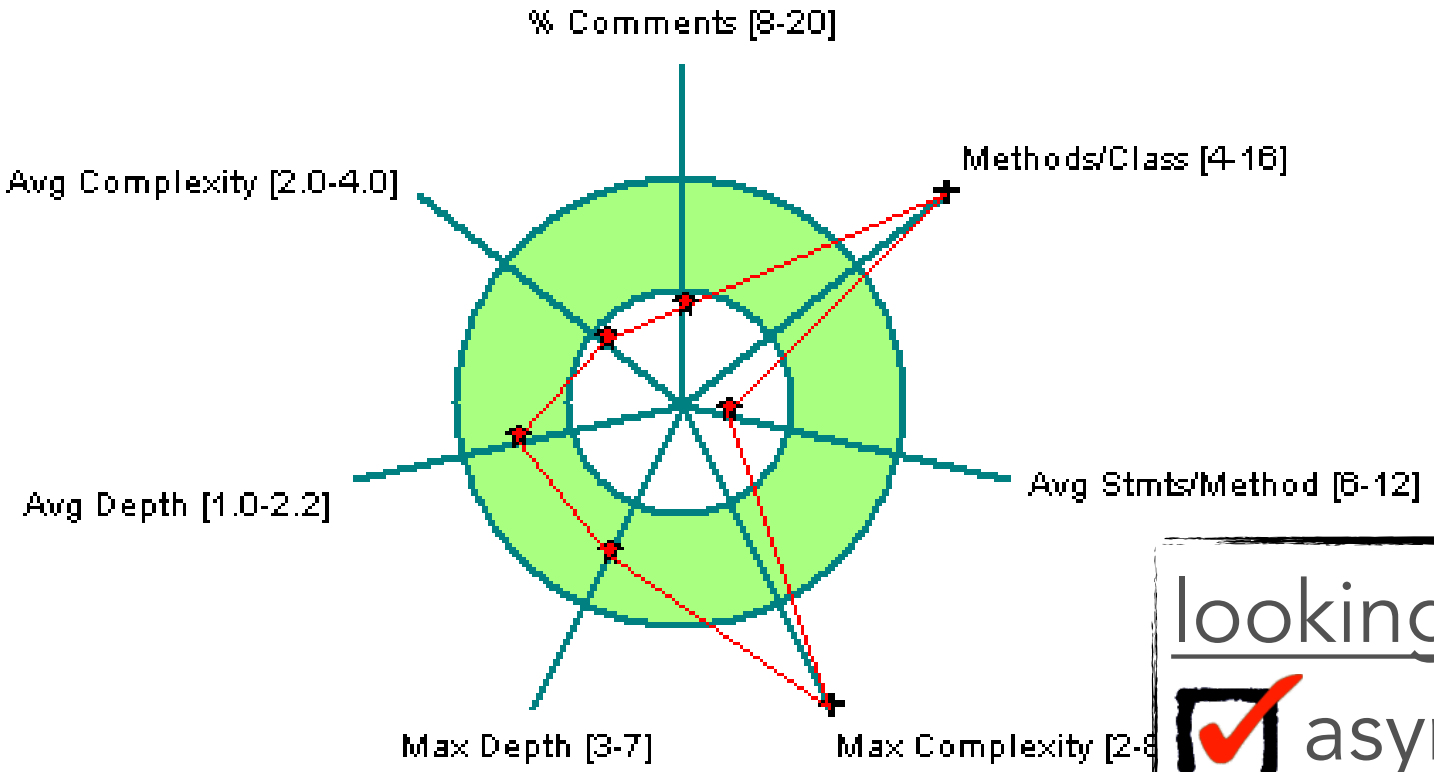
WMC	weighted methods/class	∑ of cyclomatic complexity
RFC	response for class	# of methods executed due to method call
LCOM	lack of cohesion	∑ of sets of methods not shared via sharing fields
CE	efferent coupling	∑ of other classes this class uses (outgoing calls)
CA	afferent coupling	∑ of how many other classes use this class (incoming calls)


visualizations
ERROR

Source Monitor

http://www.campwoodsw.com/sourcemonitor.html

Kiviat Metrics Graph: Project 'core'
Checkpoint 'Baseline'
File 'src\main\java\org\apache\struts2\components\DoubleListUIBean.java'



looking for:

- ☑ asymmetries
- ☑ outliers

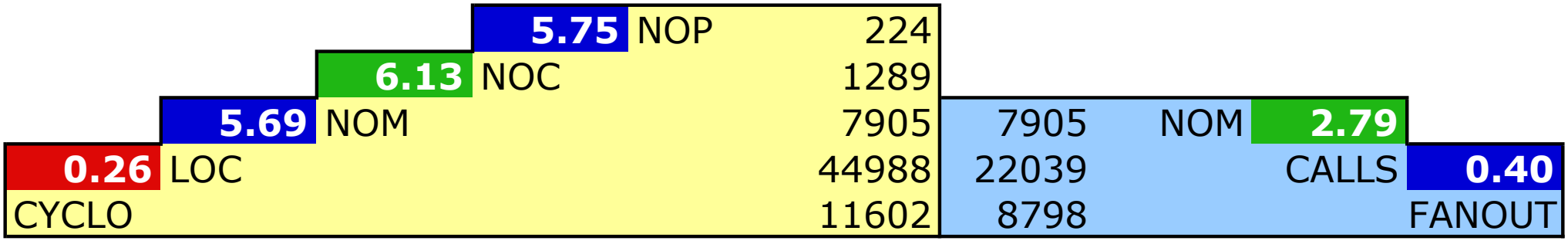

interactive-complexity-treemap.svg
CruiseControl Complexity
Details
(Click on a rectangle to view details)
Project:
Package:
File:
Class:
NCSS:
Method Coverage:
Block Coverage:
Line Coverage:
Method:
NCSS:
CCN:
Block Coverage:
Line Coverage:
CCN 1-5
CCN 6-9
CCN 10-24
CCN 25+
N/A


interactive-complexity-treemap.svg
CruiseControl Complexity
Details
(Click on a rectangle to view details)
Project: CruiseControl
Package: net.sourceforge.cruisecontrol.publishers
File: EmailPublisher.java
Class: EmailPublisher
NCSS: 345
Method Coverage: 72.0% (36.0/50.0)
Block Coverage: 67.4% (819.0/1215.0)
Line Coverage: 68.2% (187.6/275.0)
Method: createUserSet(XMLLogHelper)
NCSS: 19
CCN: 11
Block Coverage: 100.0% (122.0/122.0)
Line Coverage: 100.0% (18.0/18.0)
CCN 1-5
CCN 6-9
CCN 10-24
CCN 25+
N/A

looking for:

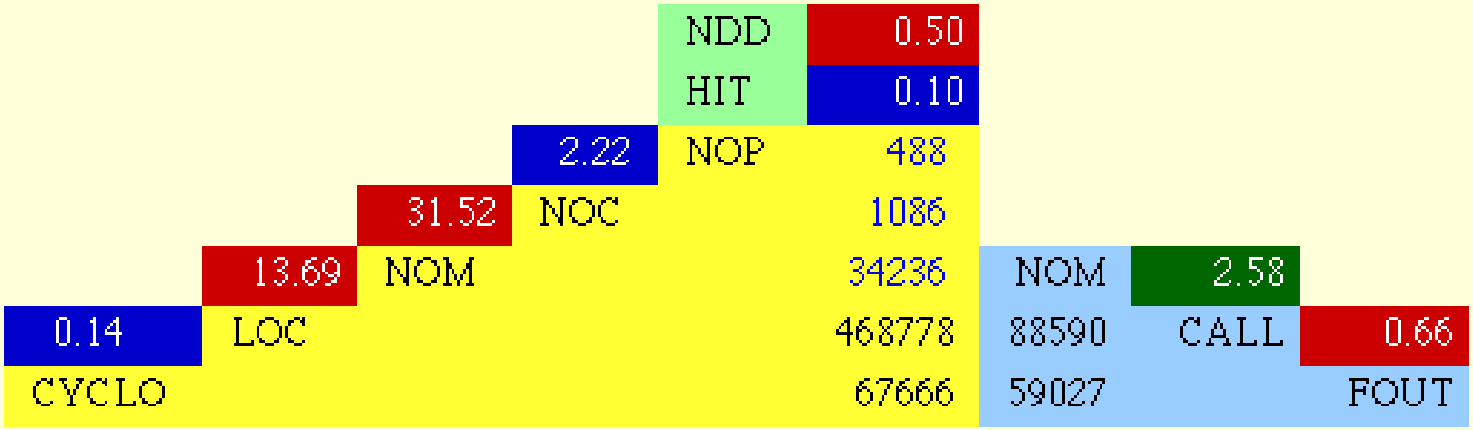
- [x] hot-spots
- [x] clusters

size & complexity pyramid



	Low	Medium	High
CYCLO / Line	0.16	0.20	0.24
LOC / method	7	10	13
NOM / class	4	7	10
NOC / package	6	17	26
CALLS / method	2.01	2.62	3.20
FANOUT / call	0.56	0.62	0.68

vuze



Interpretation of the Overview Pyramid for module
/Users/nealford/Downloads/Applications/Vuze_4.2.0.2_source.zip Folder

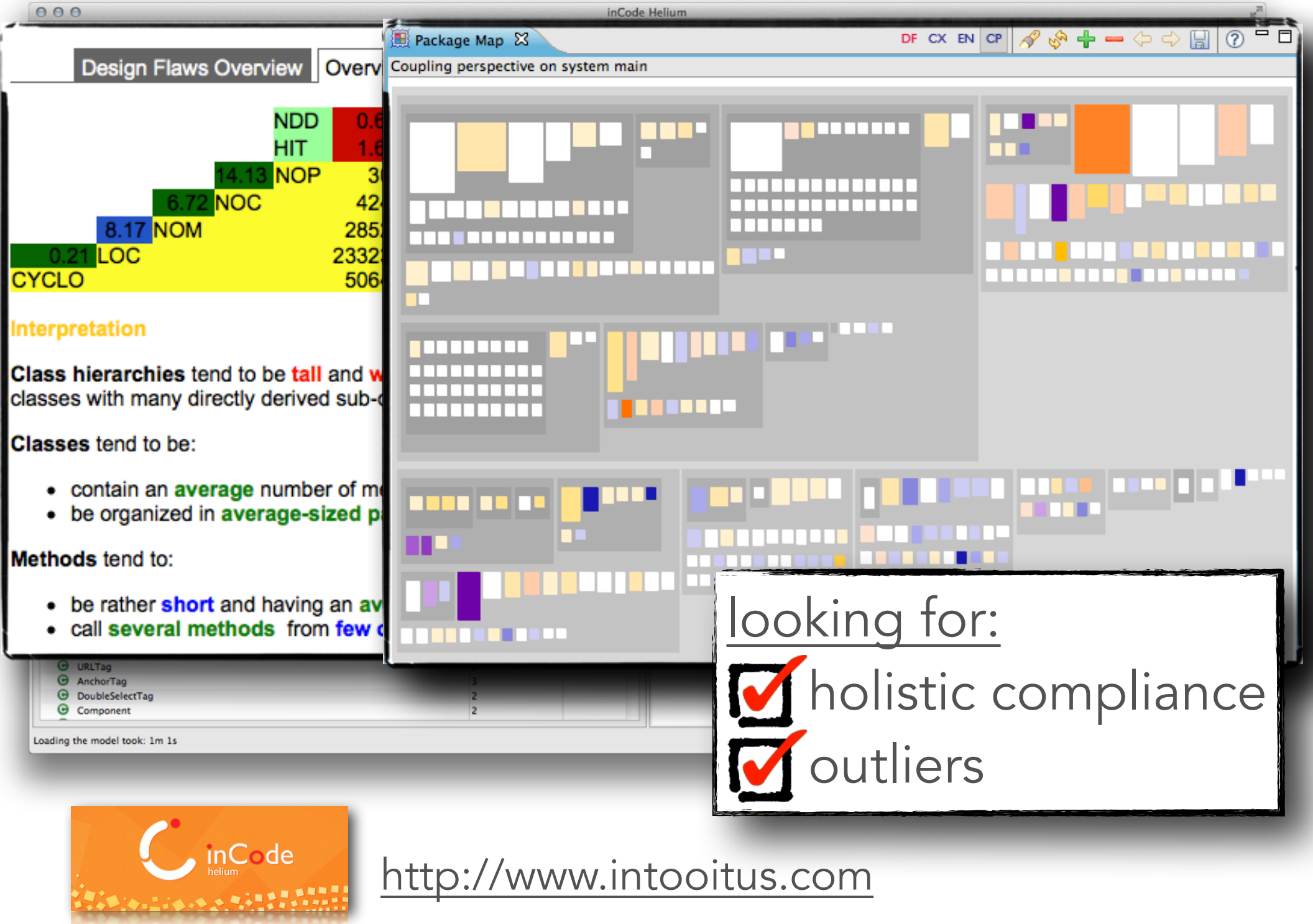
Class Hierarchies tend to be **shallow** and **wide**
(i.e. inheritance trees tend to have only few depth-level(s) and base-classes with many directly derived sub-classes)

Classes tend to:

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

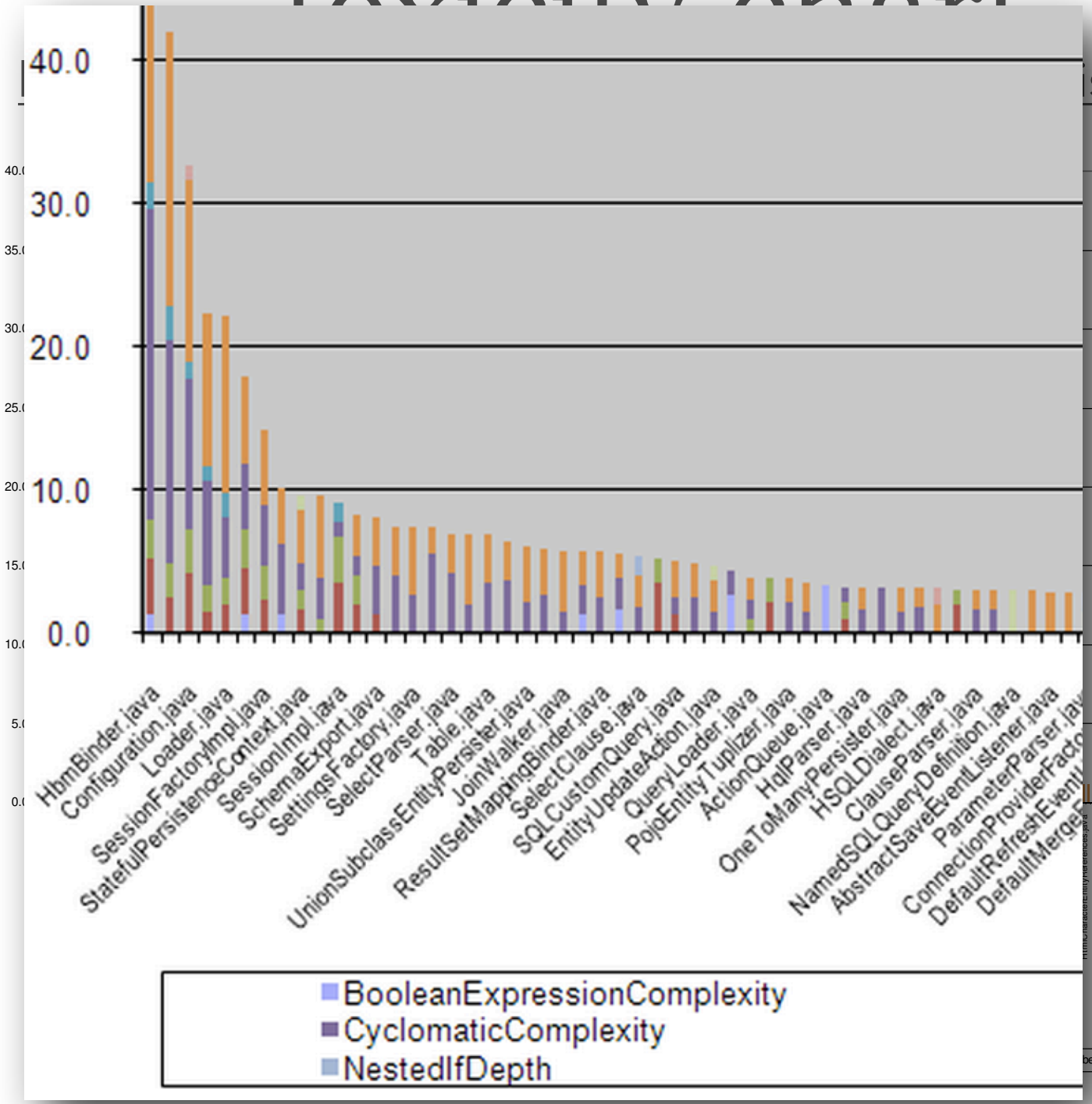
Methods tend to:

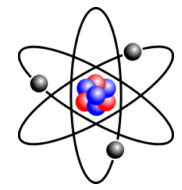
- tend to be rather **long** yet having a rather **simple logic** (i.e. few conditional branches);
- tend to call an **several methods** from **many other classes** (high coupling dispersion);


inCode Helium
Design Flaws Overview
Overv
NDD 0.6
HIT 1.6
14.13 NOP 3
6.72 NOC 42
8.17 NOM 285
0.21 LOC 2332
CYCLO 506
Interpretation
Class hierarchies tend to be tall and w
classes with many directly derived sub-
Classes tend to be:
contain an average number of m
be organized in average-sized p
Methods tend to:
be rather short and having an av
call several methods from few
Package Map
DF CX EN CP
Coupling perspective on system main
URLTag
AnchorTag 3
DoubleSelectTag 2
Component 2
Loading the model took: 1m 1s
looking for:
holistic compliance
outliers
inCode helium
http://www.intooitus.com

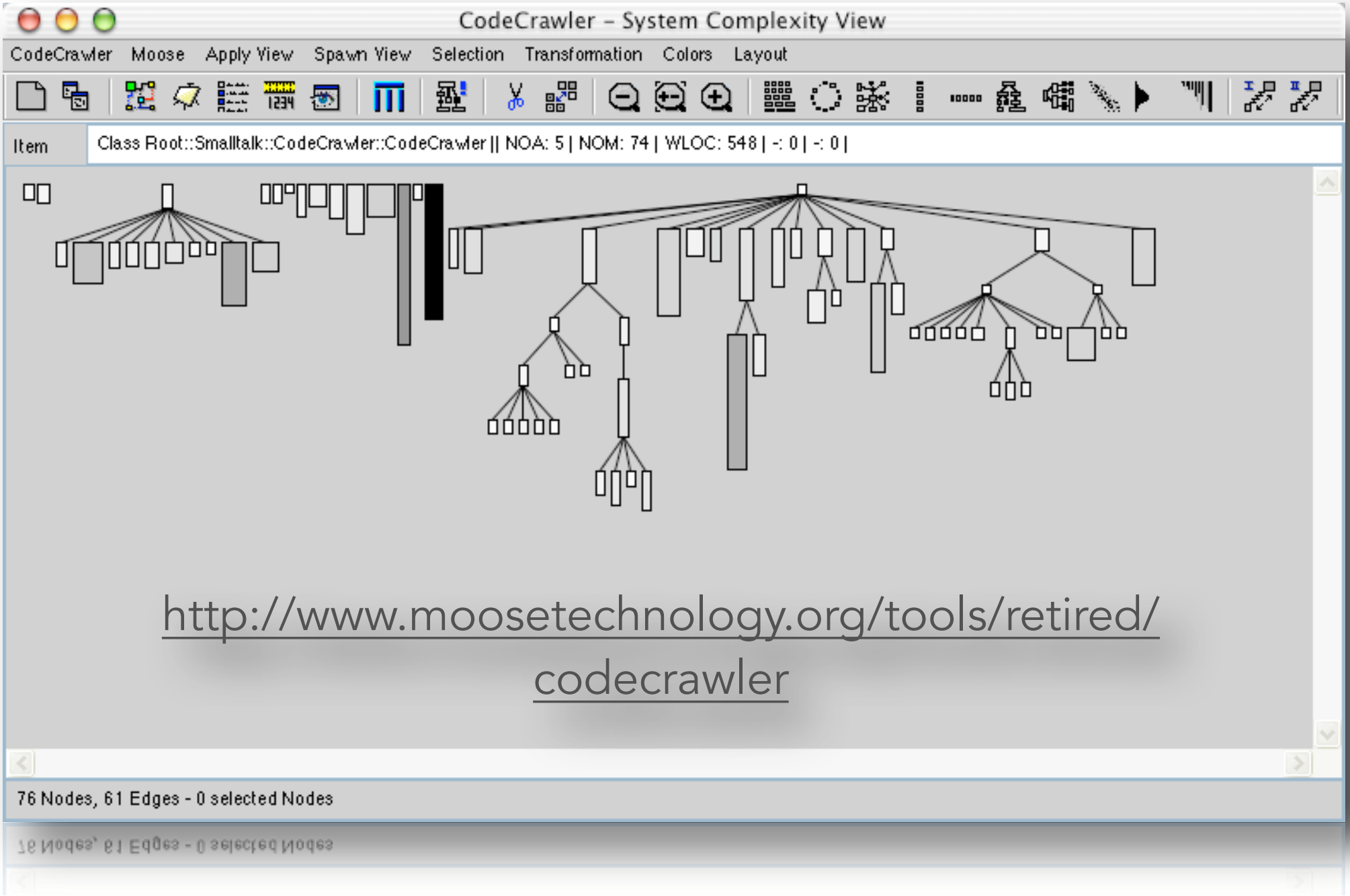
toxicity chart

s-your-code/



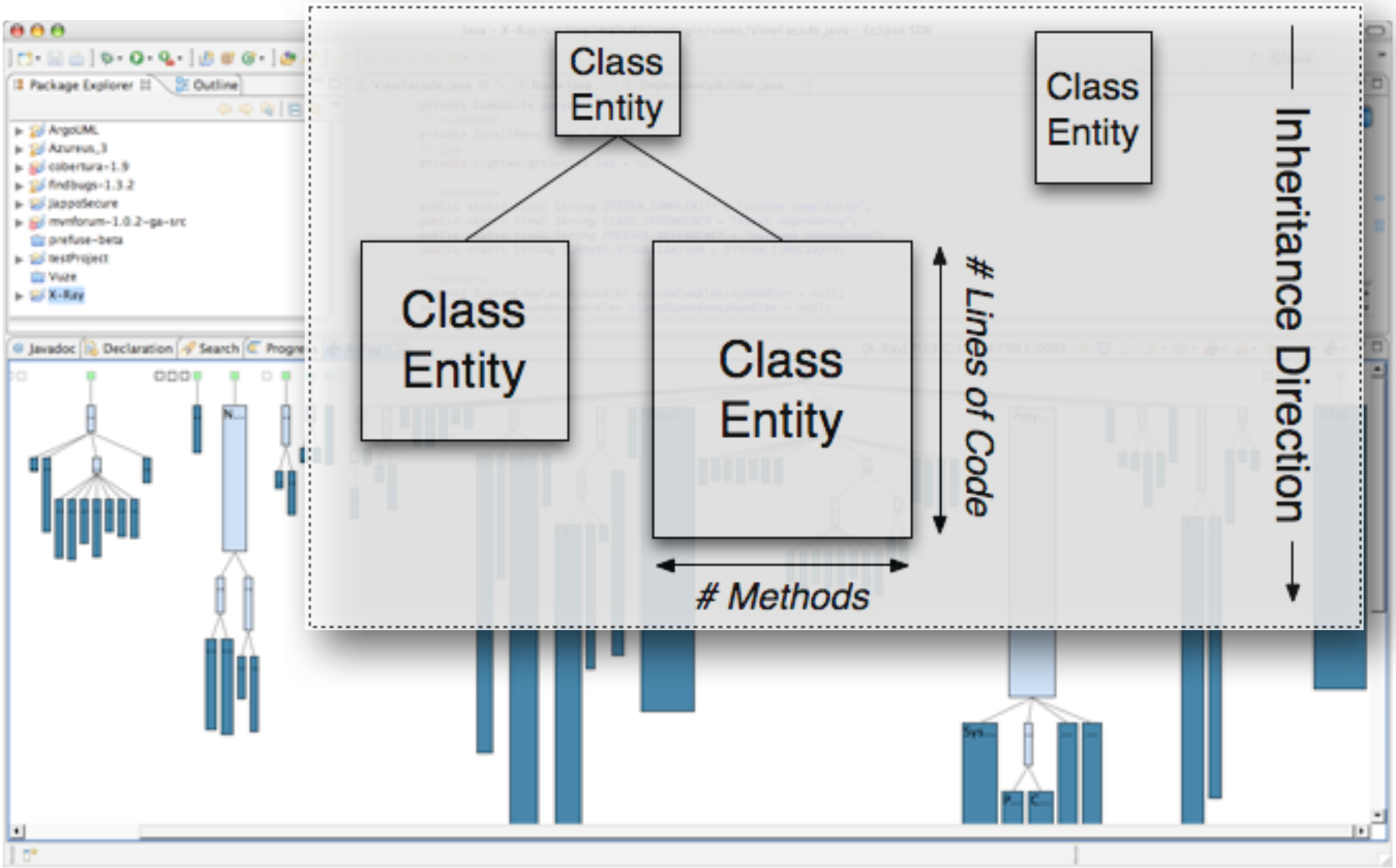


code crawler

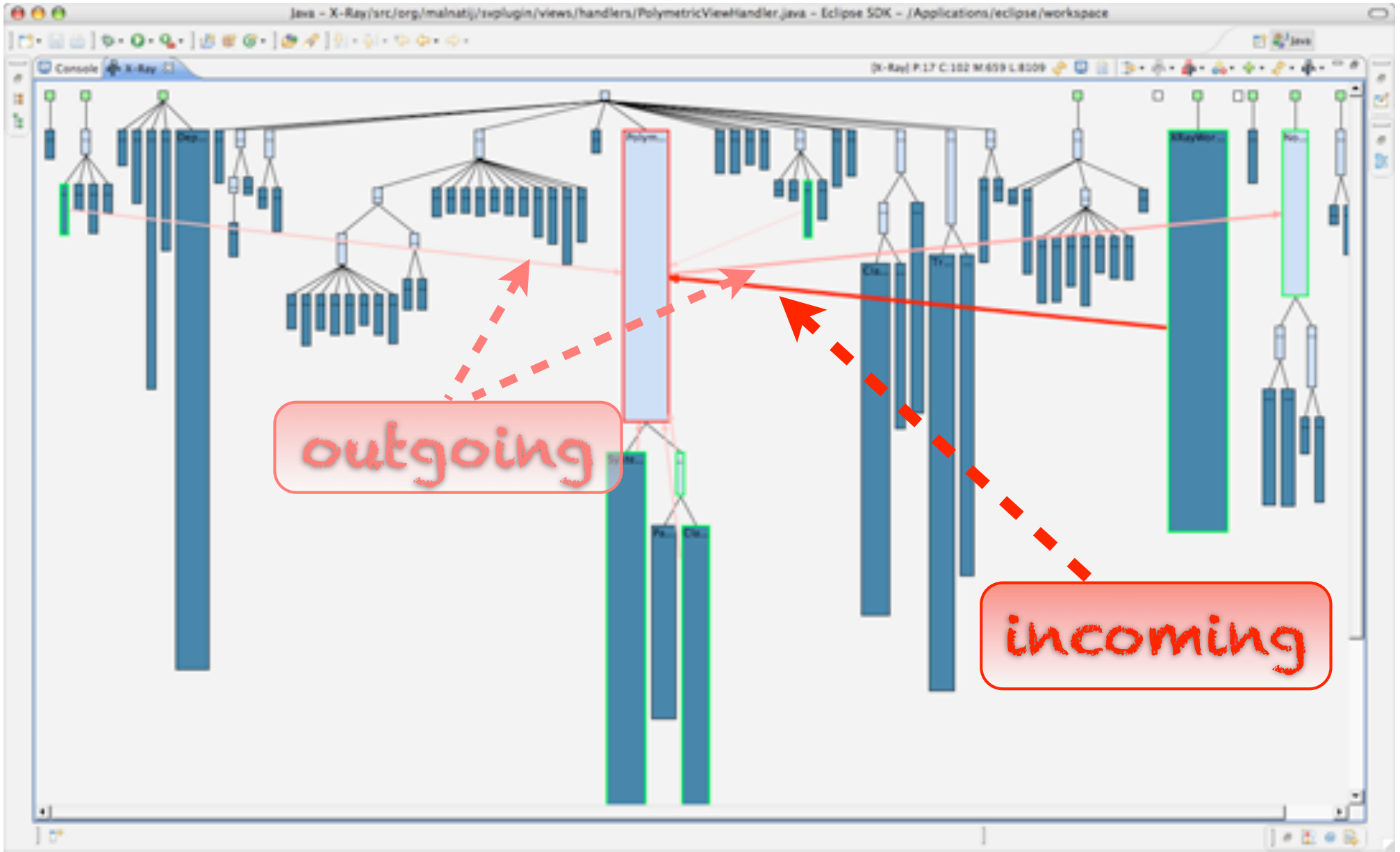


X-Ray

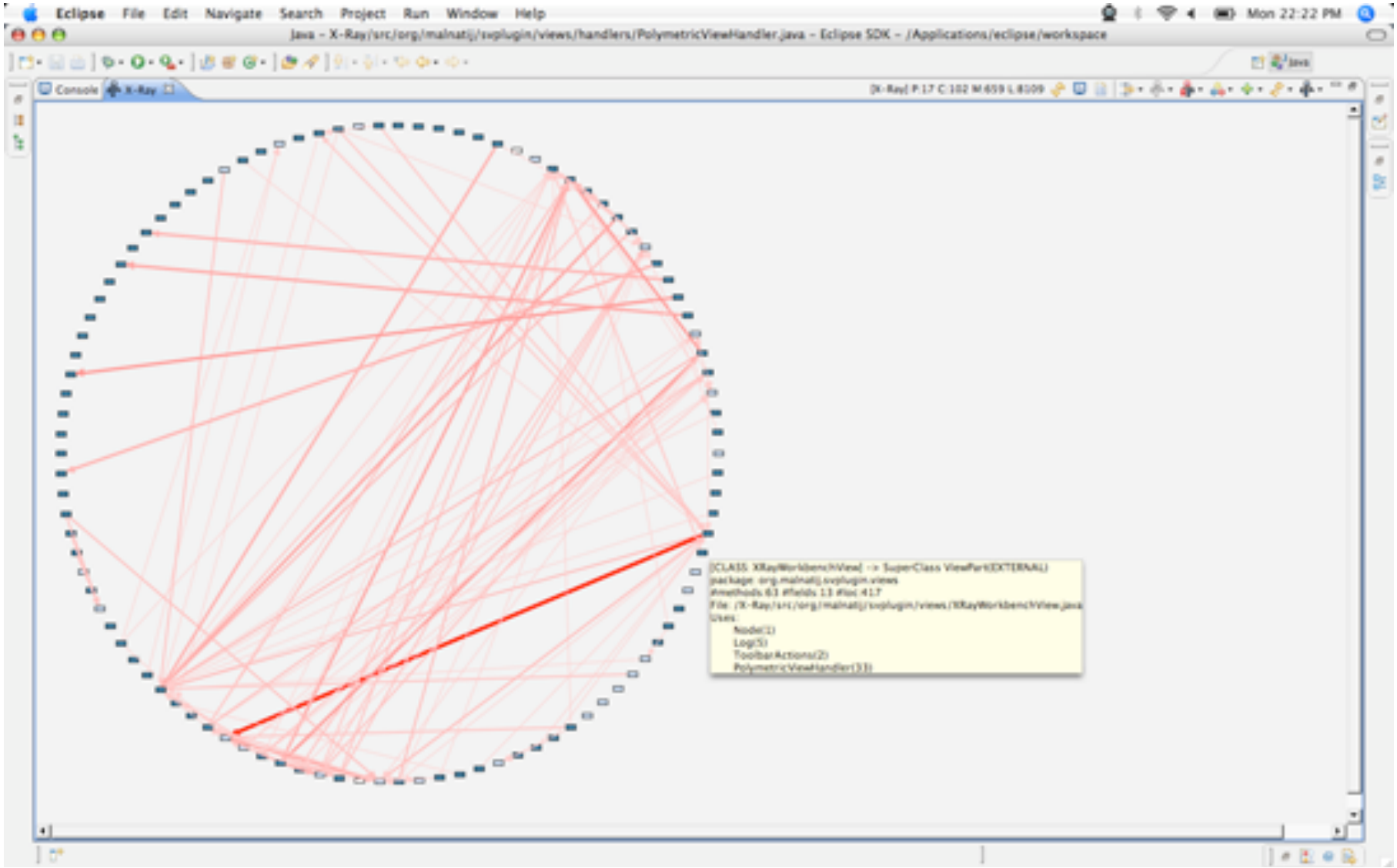
http://xray.inf.usi.ch/xray.php



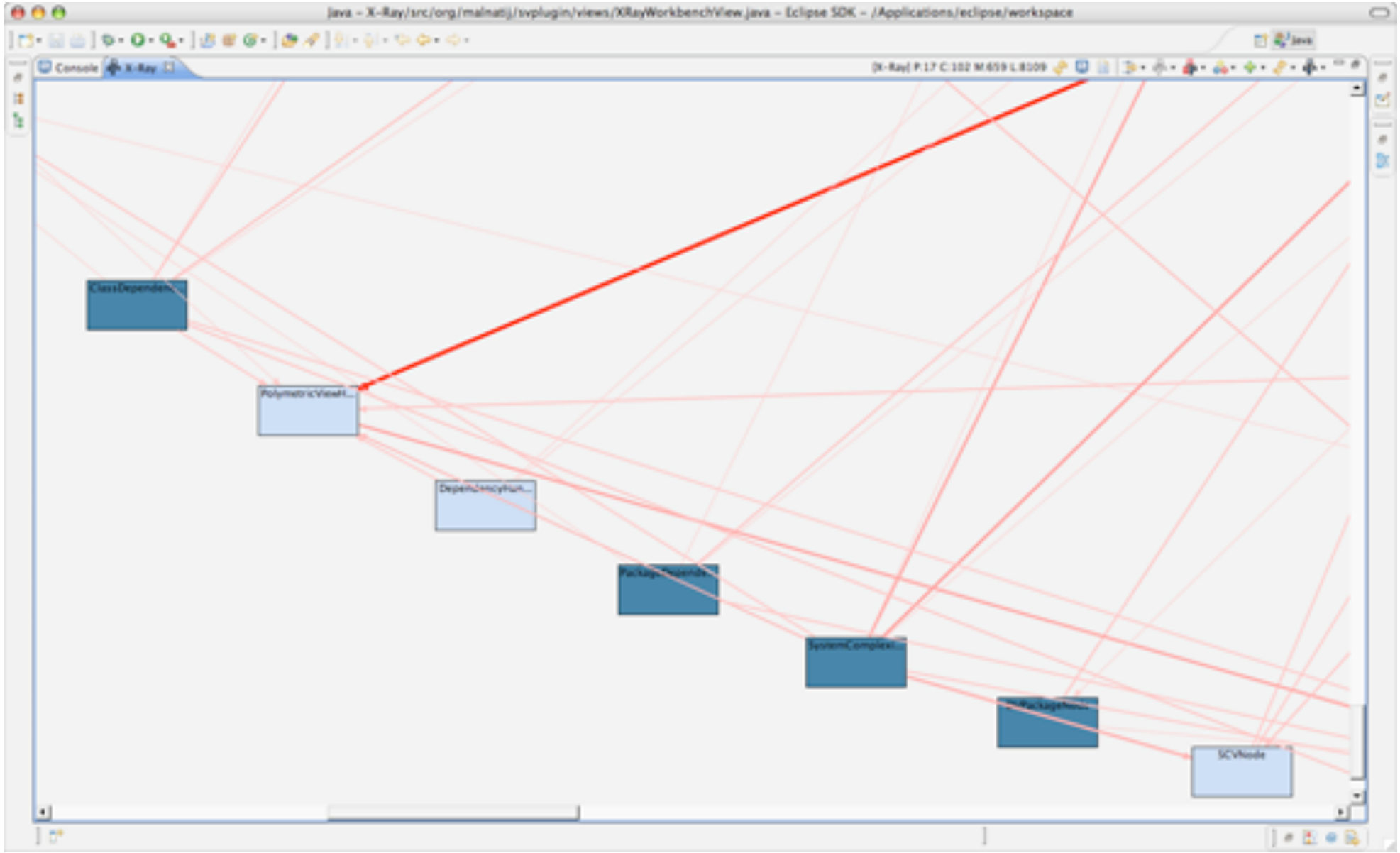
XRay dependencies



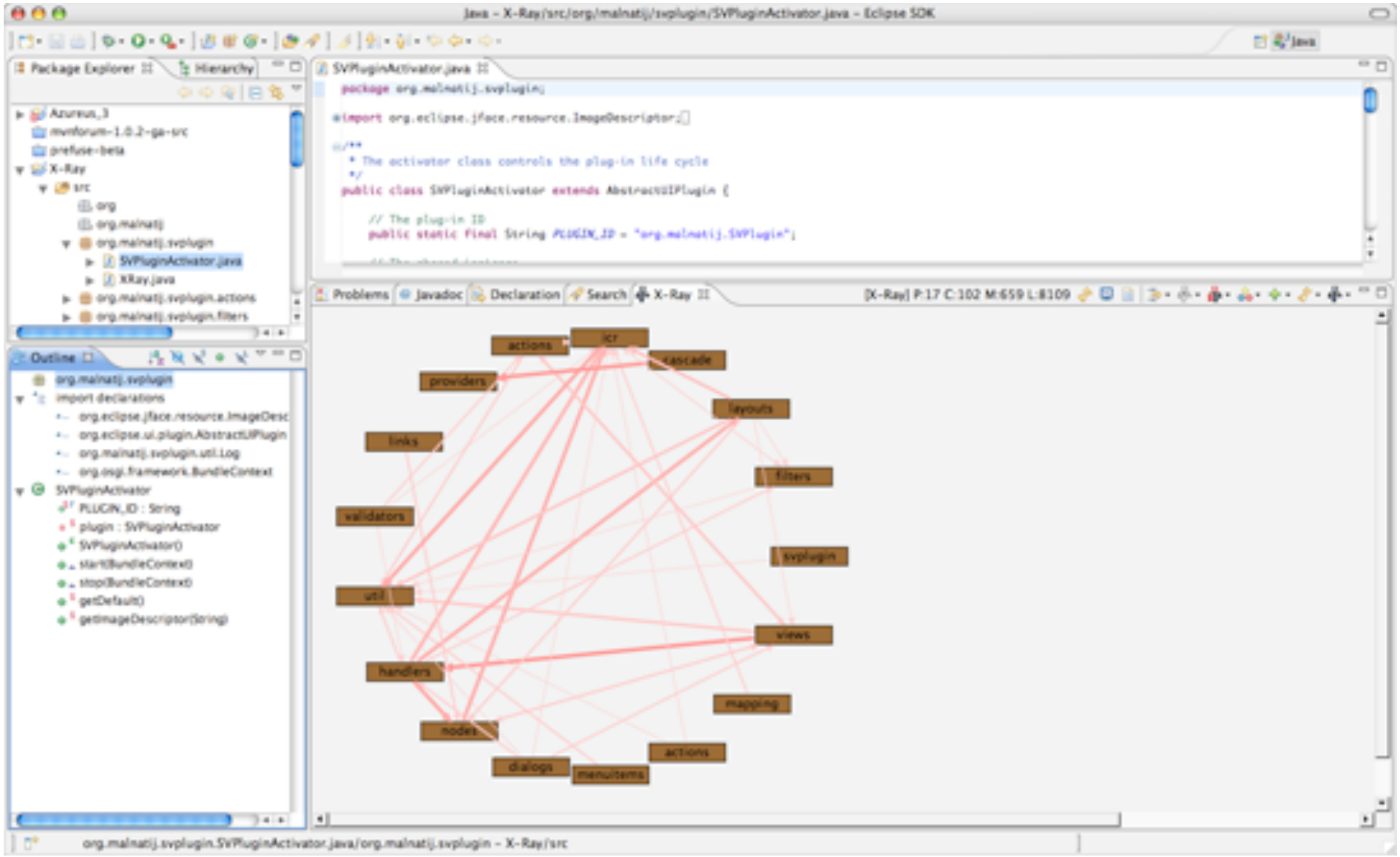
class & package dependency views



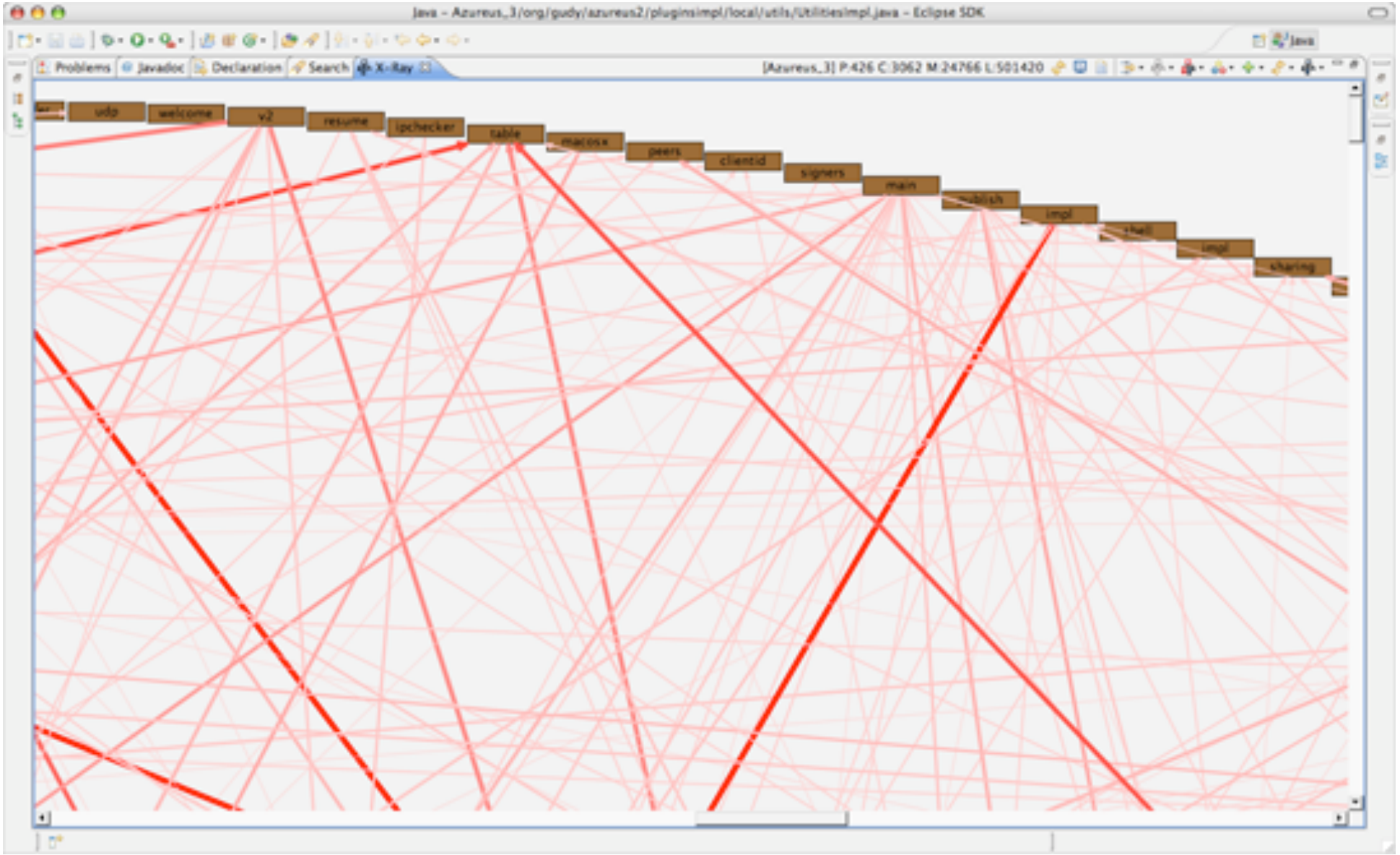
class dependency view



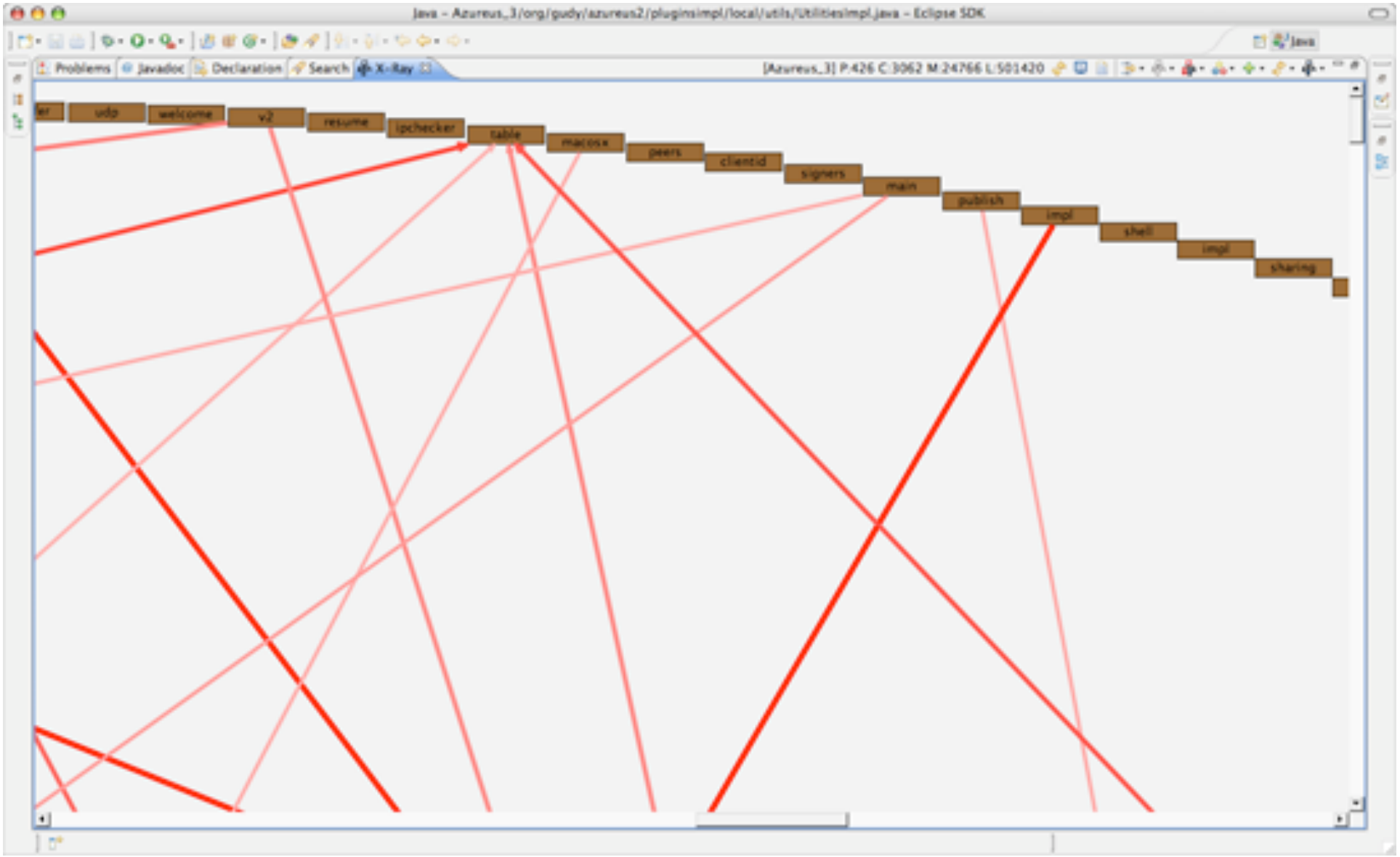
package dependency



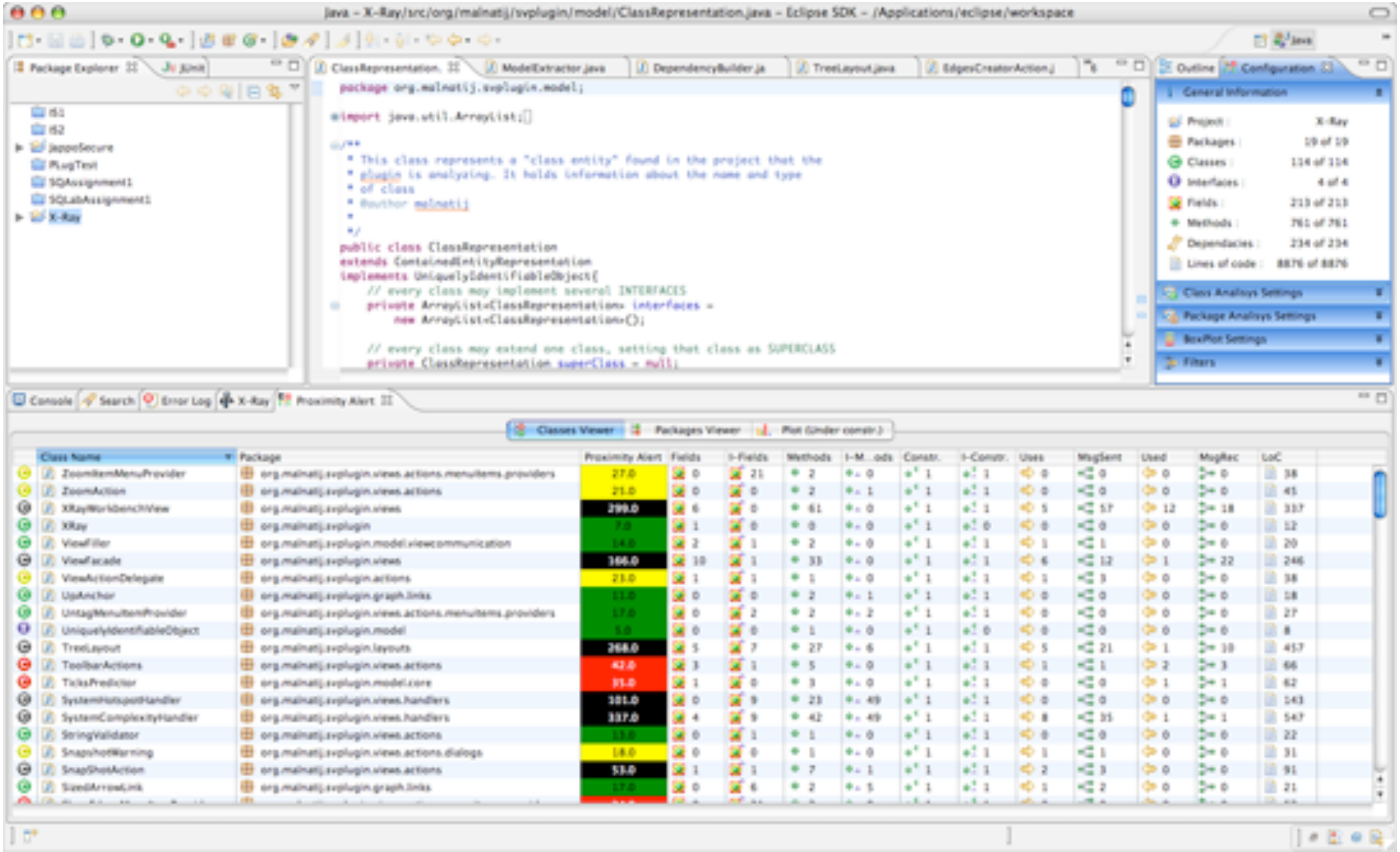
Azureus packages



filtering (< 30 weight)

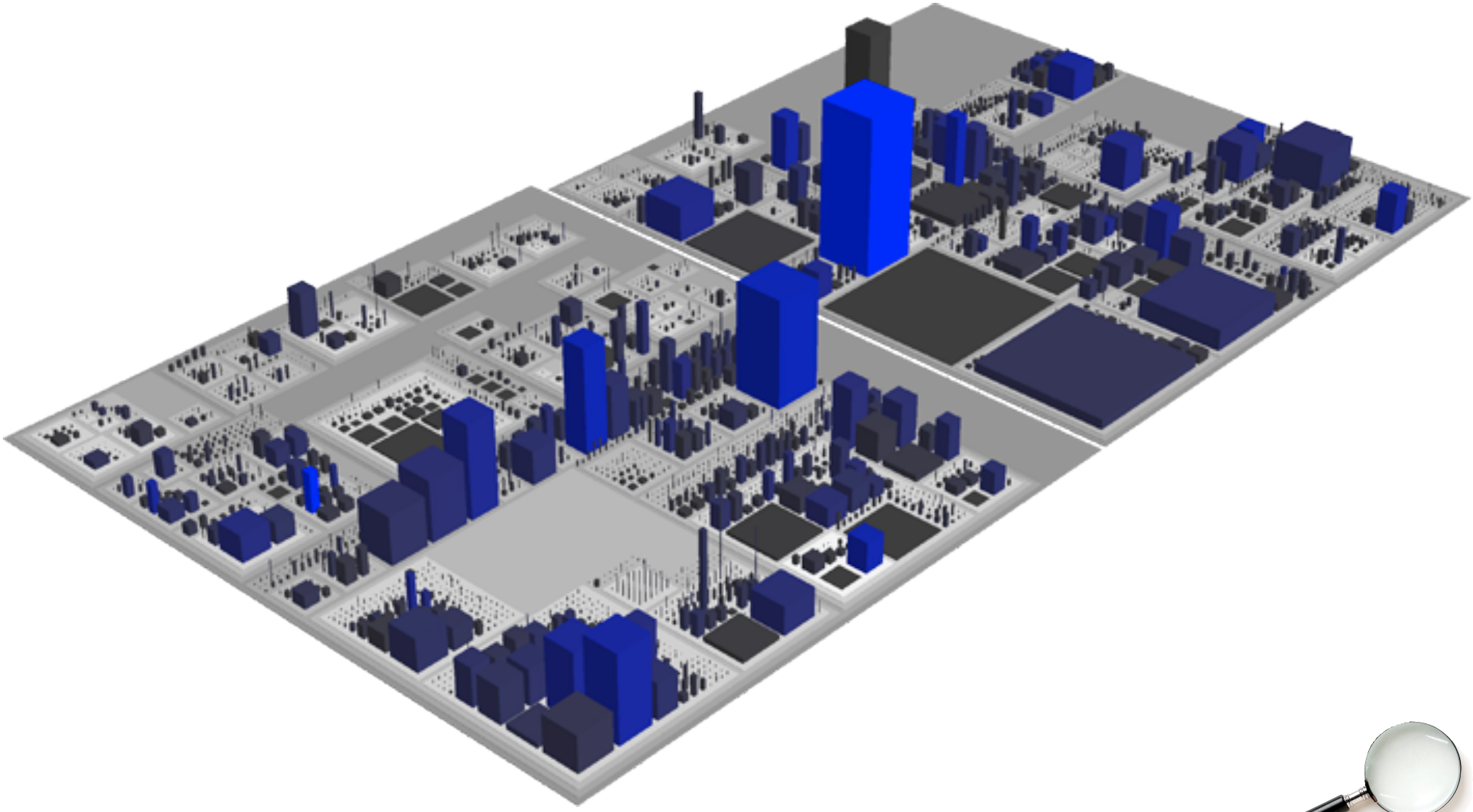


proximity alert



Code City

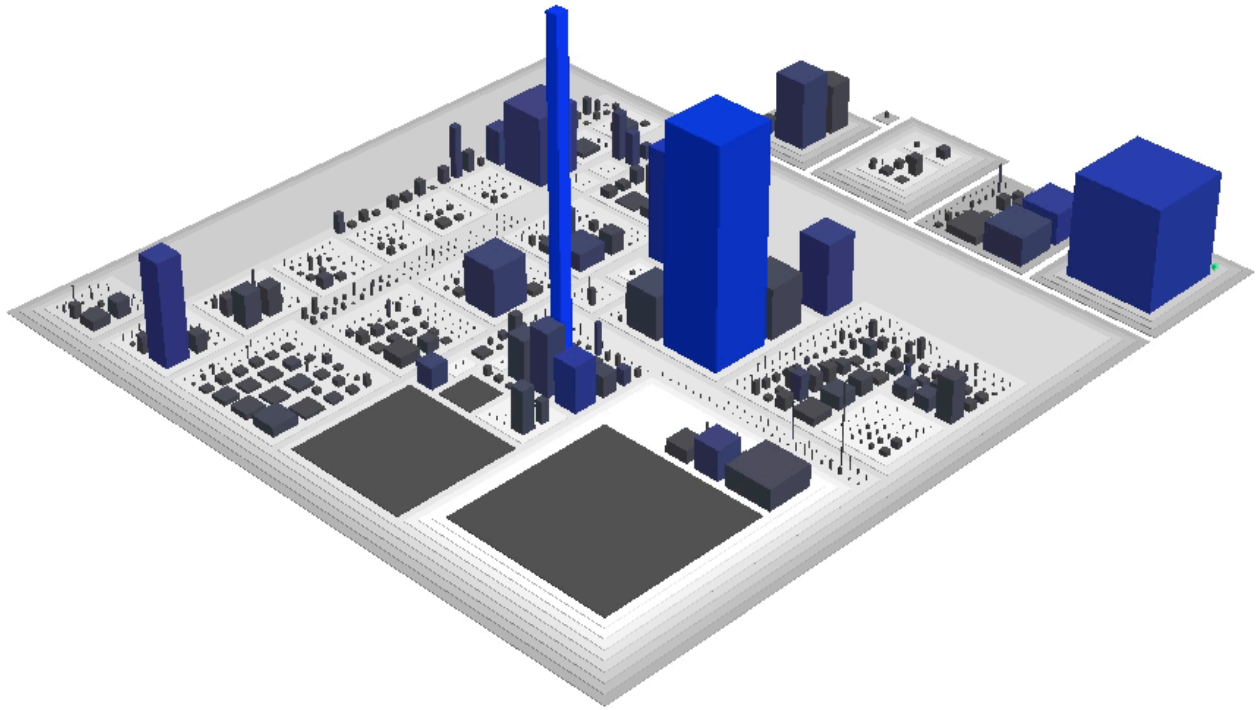
http://www.inf.usi.ch/phd/wettel/codecity.html



CodeCity

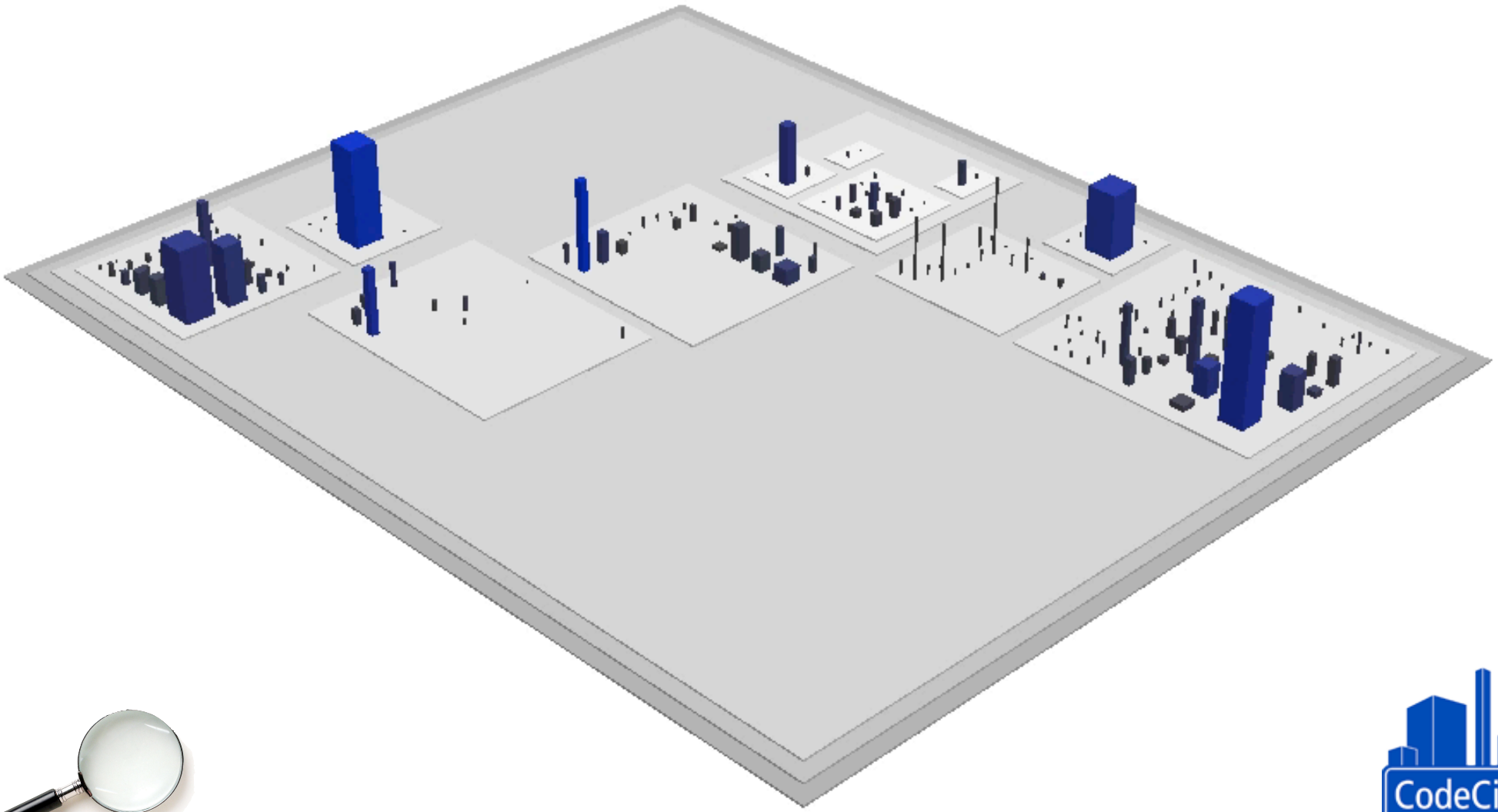
Select... On selection... Help

Selected: class XmlException

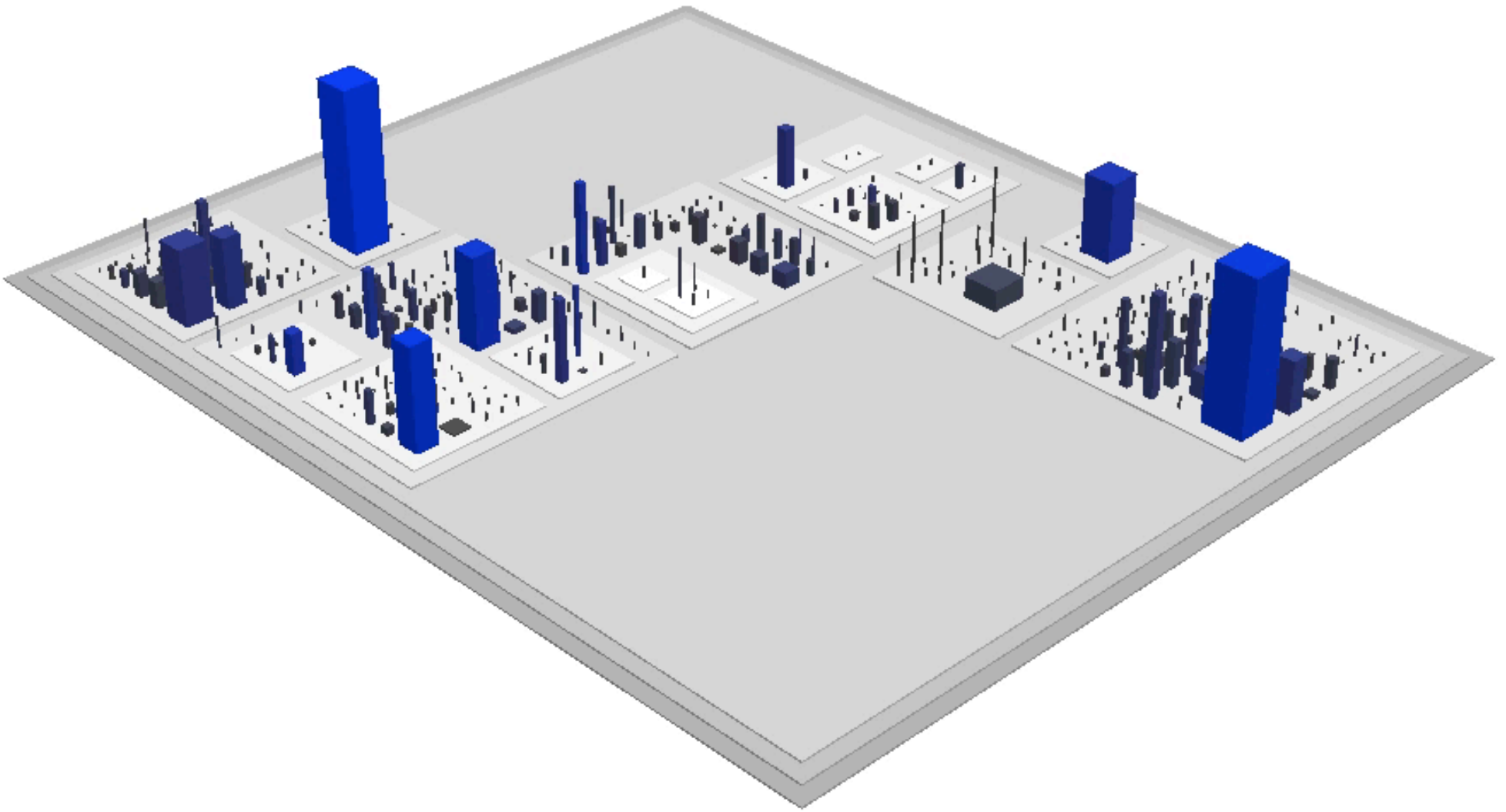


- com
  - com::microstar
    - com::microstar::xml
      - HandlerBase
      - XmlHandler
      - XmlException
      - XmlParser
  - de
  - org
  - doclet
  - installer
  - macosx
  - net

time sequence



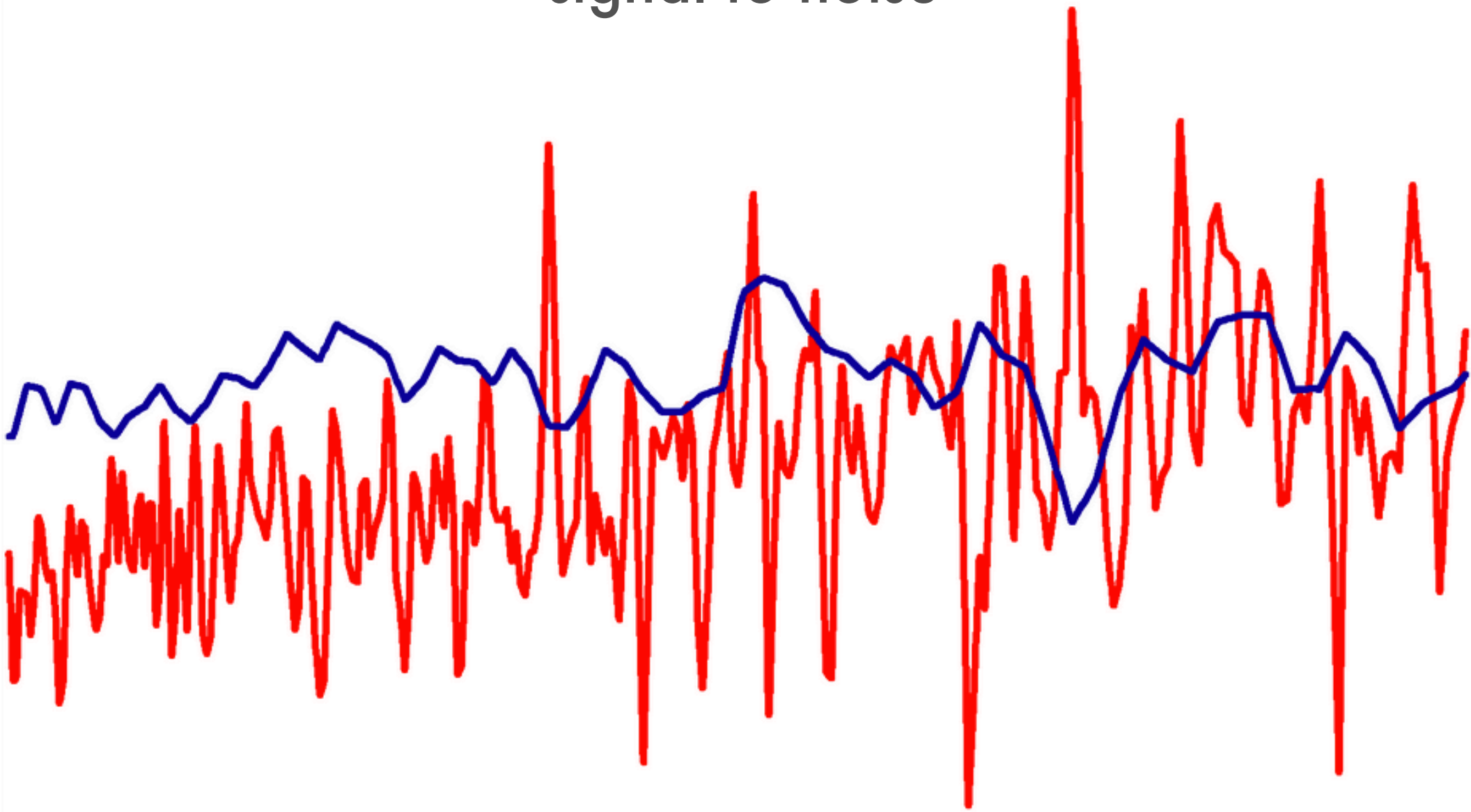
Select... On selection... Evolution Select classes where Help



limitations of metrics

signal to noise

signal to noise



limitations of metrics

signal to noise

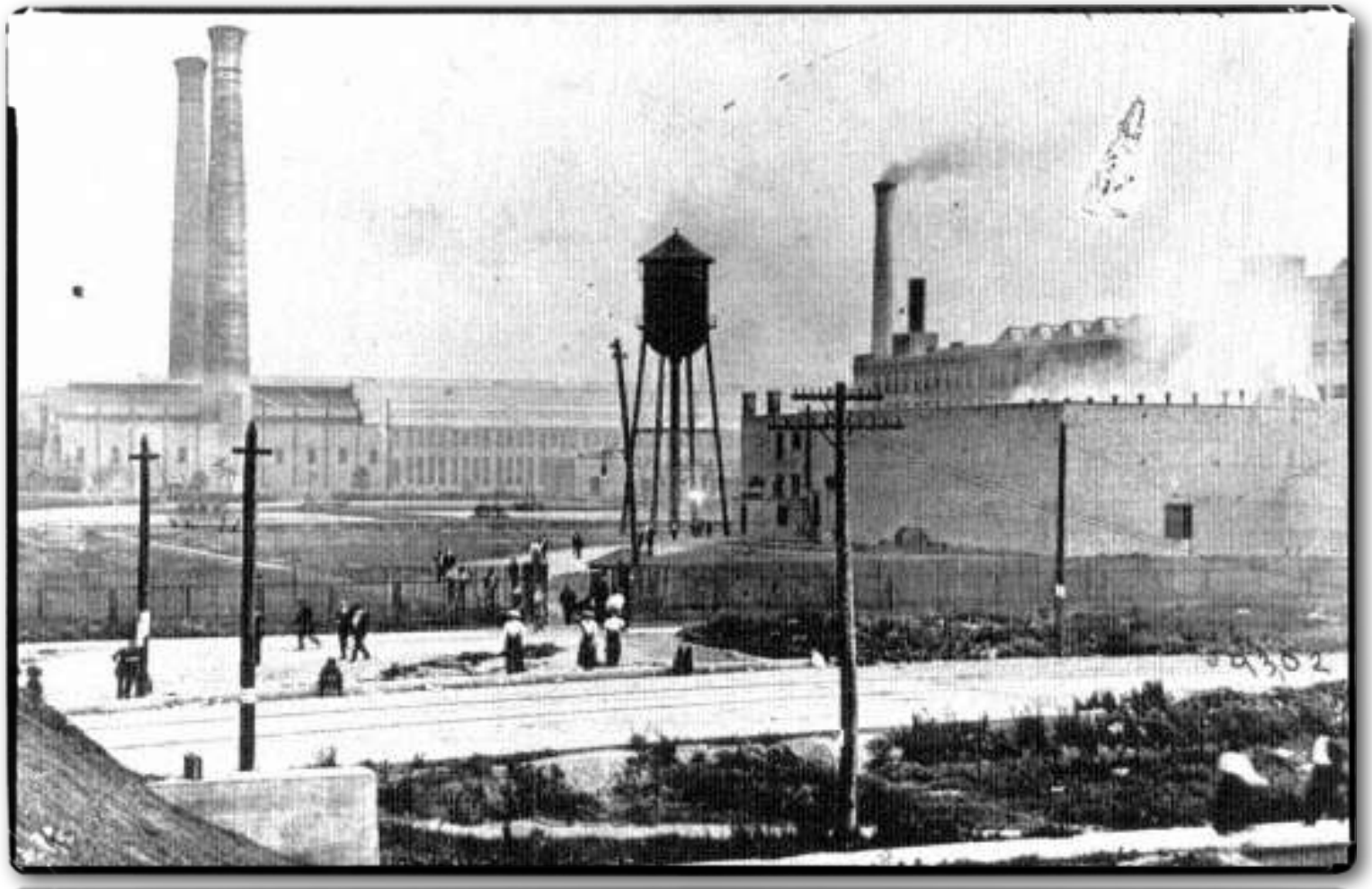
no “1 True Metric”

gathered but ignored

inaction / overaction

prefer trends to discrete values

the Hawthorne Effect



measure & let it be known that you are measuring

limitations of metrics

signal to noise

no “1 True Metric”

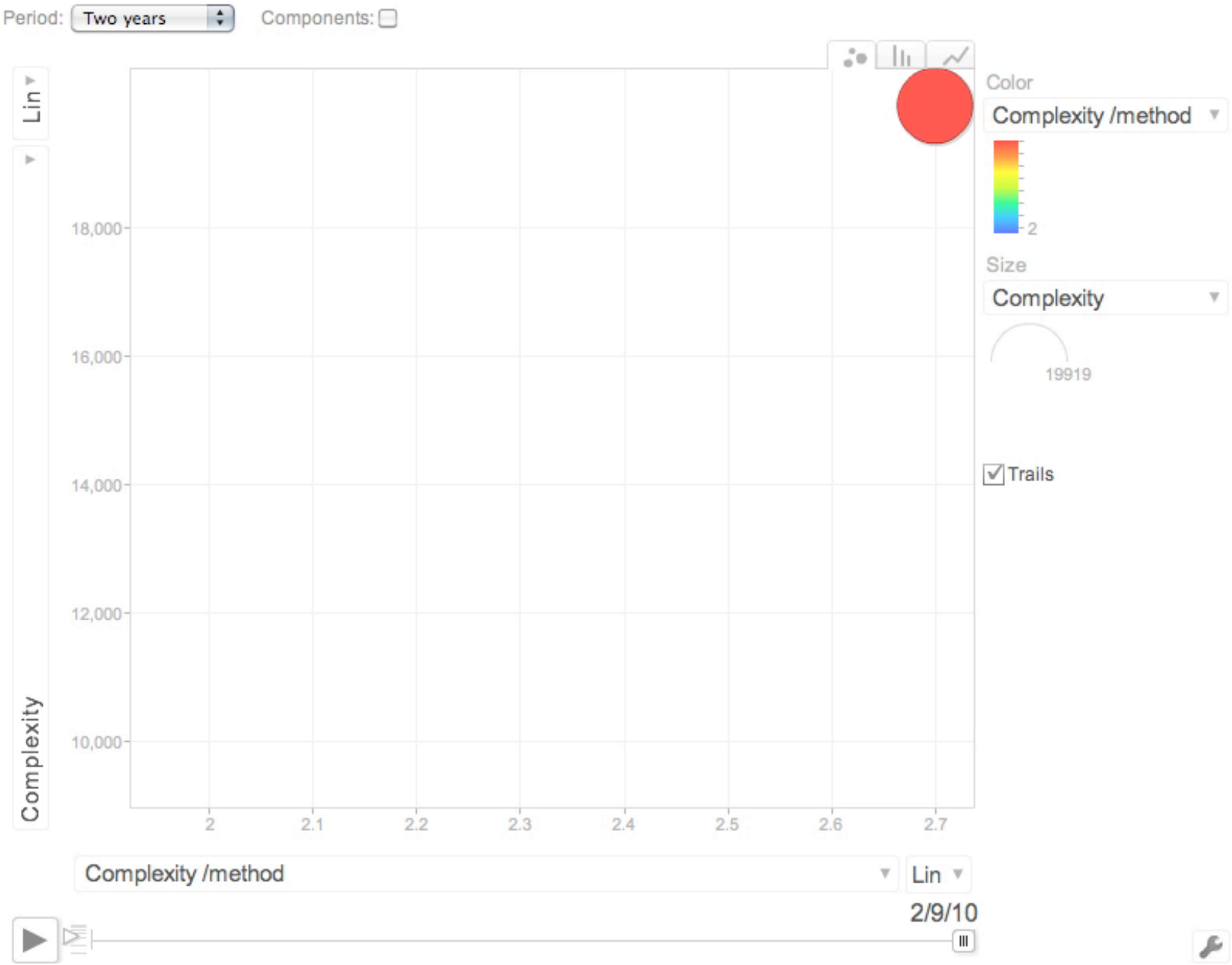
gathered but ignored

inaction / overaction

prefer trends to discrete values

Sonar trends

http://www.sonarsource.org/



2 typical uses for metrics

radiators

probes


https://github.com/thehammer/jukebox.rb

?'S

Mark Richards
Independent Consultant
Hands-on Enterprise / Integration Architect
Published Author / Conference Speaker

http://www.wmrichards.com
http://www.linkedin.com/pub/mark-richards/0/121/5b9

Published Books:
Java Message Service, 2nd Edition
97 Things Every Software Architect Should Know
Java Transaction Design Strategies

Neal Ford
Director / Software Architect /
Meme Wrangler

ThoughtWorks®

2002 Summit Blvd, Level 3, Atlanta, GA 30319, USA
T: +1 40 4242 9929 Twitter: @neal4d
E: nford@thoughtworks.com W: thoughtworks.com