

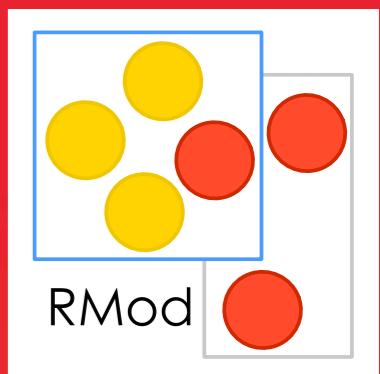
Dedicated tools and research for Software Business intelligence

@ Tisoca 2014

Dr. S. Ducasse

rmod.lille.inria.fr

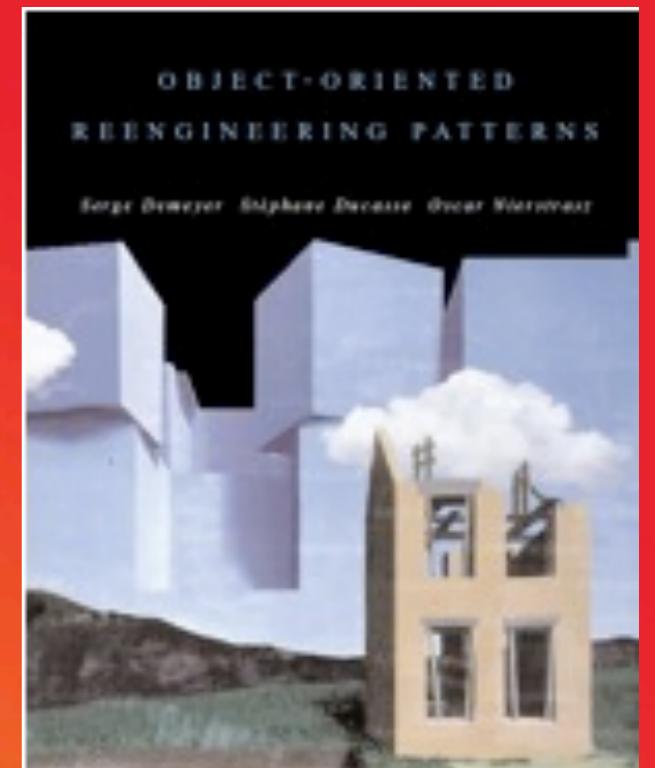
stephane.ducasse.free.fr



synectique
Inventive Analysis

stephane.ducasse.free.fr

- Leader of RMOD Inria team
- Co-founder of <http://www.synectique.eu>
- Co-creator of Moose <http://www.moosetechnology.org>
- Leader of <http://www.pharo.org>
- Coder and object-designer
- Traits co-father



A word cloud visualization centered around software development and related topics. The words are arranged in three main vertical columns. The left column contains terms like 'Programming', 'Development', 'Duplicated', 'Visualization', 'Extensions', and 'Feature Cycles'. The middle column contains 'Software', 'Reflective', 'Object analysis', 'Information', 'Repositories', 'Code', 'Understanding', 'IDE', 'Class', 'Scripting', 'History', 'Maps', 'Meta', 'Remodularisation', 'Mining', 'Design', 'Web', 'Metrics', 'Array', and 'Rules'. The right column contains 'Pharo', 'Evolution', 'Layers', 'Tools', 'Identification', 'Moose', 'Traits', 'Visual', 'Dependency', 'Encapsulation', 'Quality', 'Reengineering', 'Models', 'Dynamic', 'Modules', 'Interexchange', 'Proxy', 'formats', 'Checking', 'Reverse', and 'Extraction'. The words are colored in shades of red, orange, and blue.

Merge Trace
Reflection
Bootstrap
Serialization
Teaching
Comprehension
Patterns
Language
Validation
Semantic
Aspects
Engineering
IDE
Class
Scripting
History
Maps
Meta
Remodularisation
Mining
Extensions
Design
Feature Cycles

Object analysis
Analyses
Information
Repositories
Code
Understanding
IDE
Scripting
History
Maps
Meta
Remodularisation
Mining
Extensions
Design
Feature Cycles

Layers
Evolution
Tools
Identification
Moose
Traits
Visual
Dependency
Encapsulation
Quality
Reengineering
Models
Dynamic
Modules
Interexchange
Proxy
formats
Checking
Reverse
Extraction

Software is

Complex

Software is a living entity...

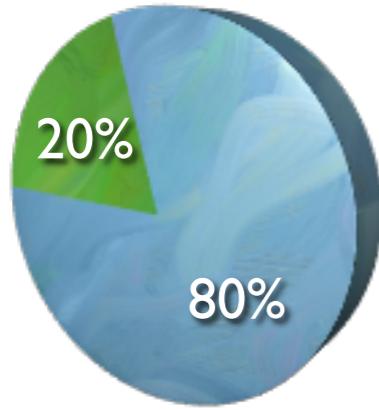
- Early decisions were certainly good at that time
- But the context changes
- Customers change
- Technology changes
- People change



Maintenance = Success!!

We only
maintain
useful
successful
software

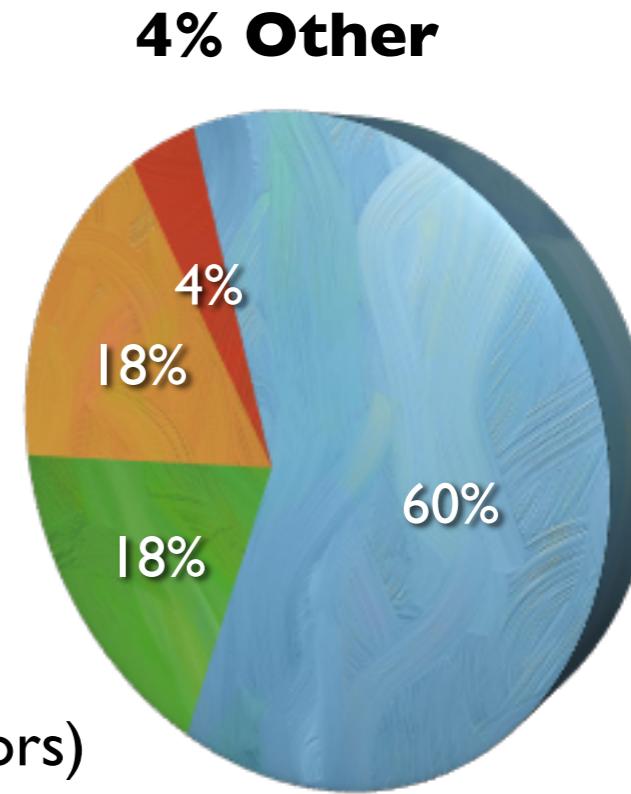
Maintenance is *continuous* Development



Between **50%** and **90%** of
global effort is spent on
“maintenance” !

18% Adaptive
(new platforms or OS)

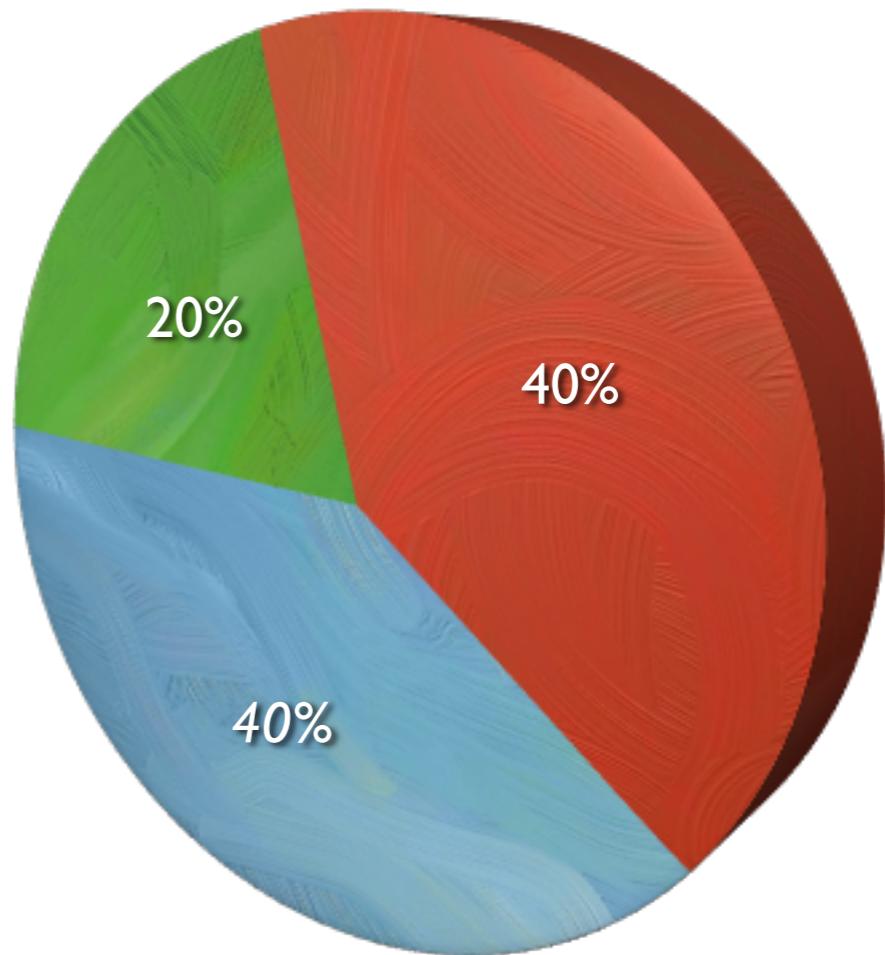
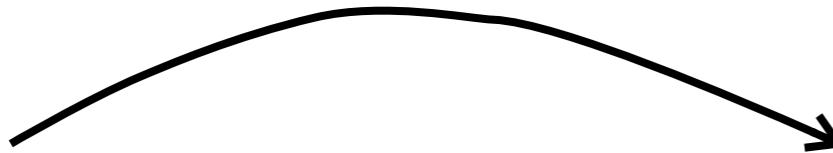
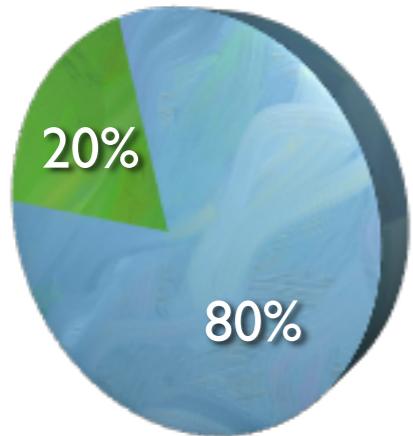
18% Corrective
(fixing reported errors)



60% Perfective
(new functionality)

“Maintenance”

50% of development time is lost trying to understand code !

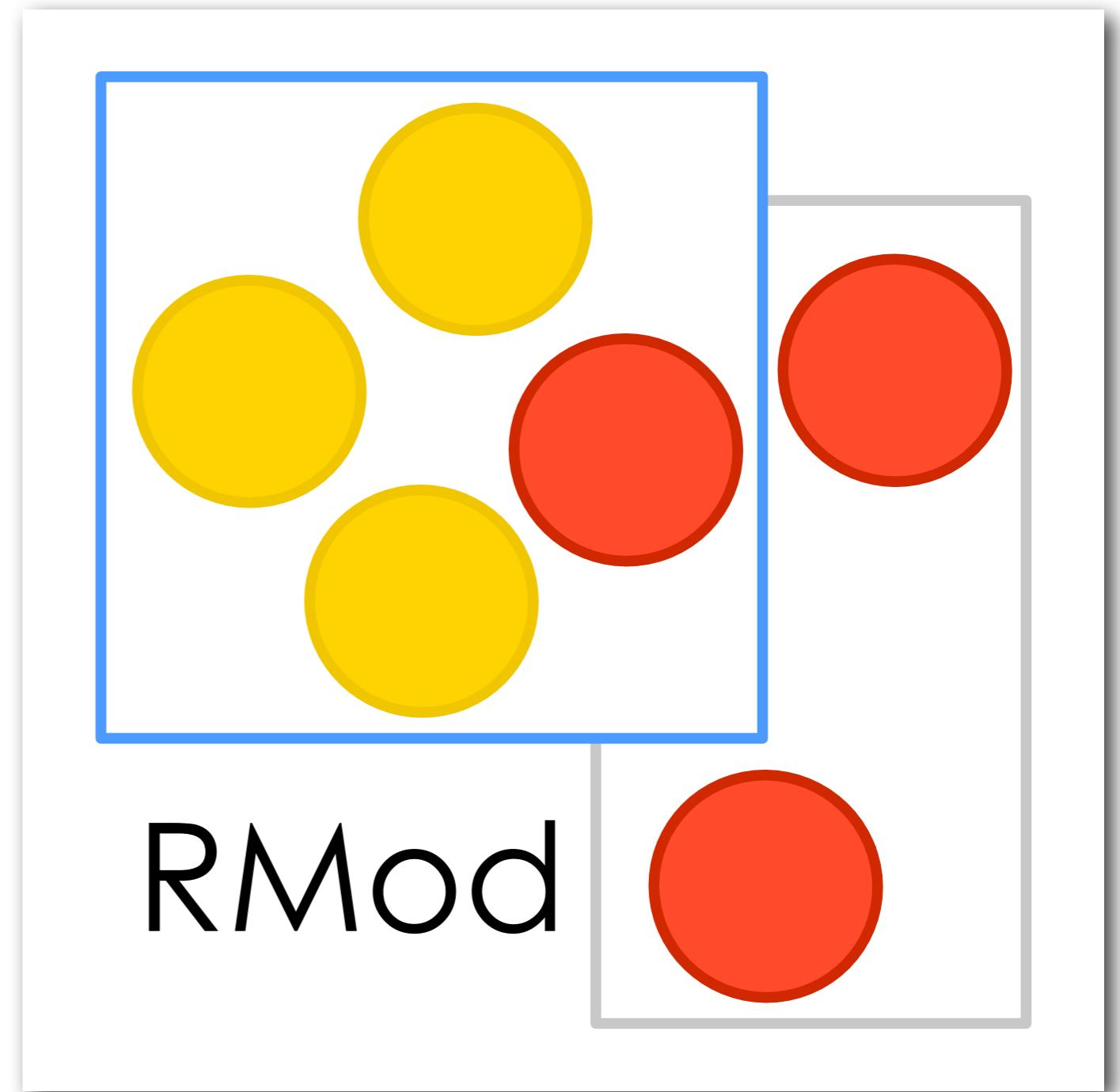


Between **50%** and **80%** of the
overall cost is spent in the evolution

We lose a lot of time with inappropriate and
ineffective practices

We need
dedicated
tools!





Pragmatic researchers in action



Dedicated tools tailored to your
problems

Profitable in terms of cost

<http://www.synectique.eu>

RMOD

RMod: code analysis, metamodeling, software metrics, program understanding, program visualization, evolution analysis, refactorings, legacy code, quality, ...

Current focus

Remodularization analyses

Quality models (PSA-Airfrance)

Towards semantic merge

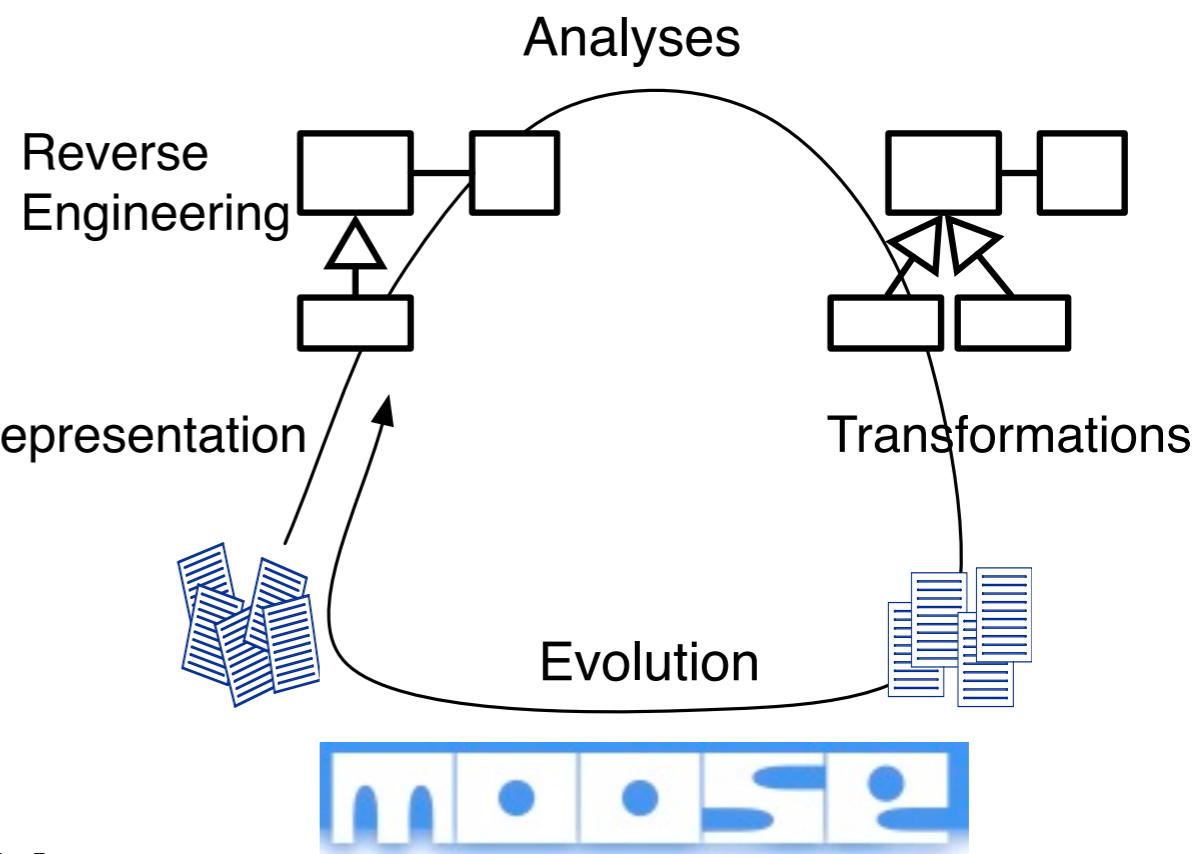
Old and odd language analyses

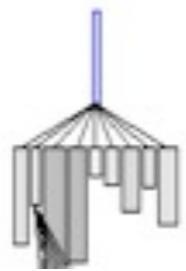
Rule and bug assessment

Collaborations

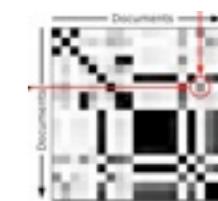
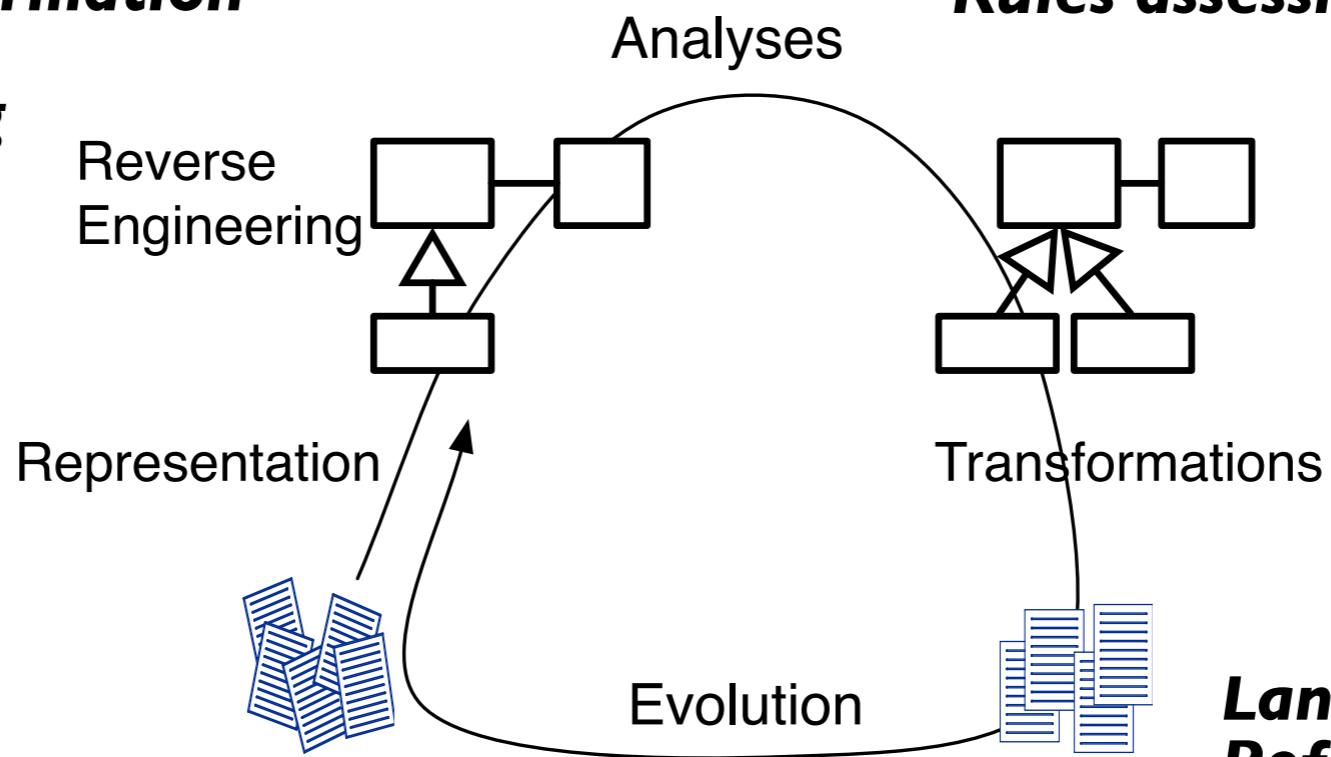
Soft-VUB (Belgium), Pleiad (Chile)

UFMG (Brazil), SCG (Swiss), LIRMM



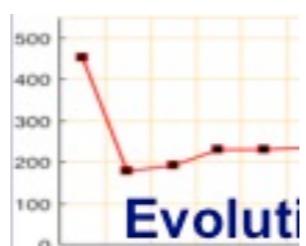


Understanding Large Systems
Static/Dynamic Information
Feature Analysis
Class Understanding
Package Blueprints
Distribution Maps

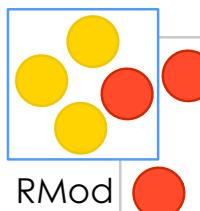
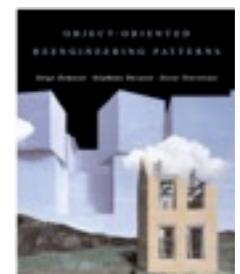
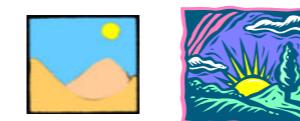


Language Independent Refactorings

Language Independent Meta Model (FAMIX)
An Extensible Reengineering Environment



Reengineering Patterns
Version Analyses
HISMO metamodel



Software Metrics
Quality Models
Duplicated Code Identification
Test Generation
Cycle and Layer Identification
Merging techniques
Rules assessment

McCabe = 21

NOM = 102
LOC = 753,000

Metrics



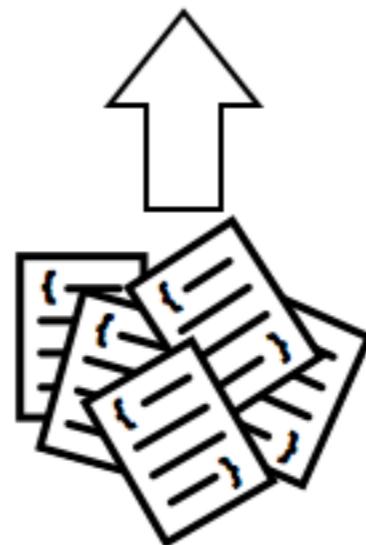
Queries



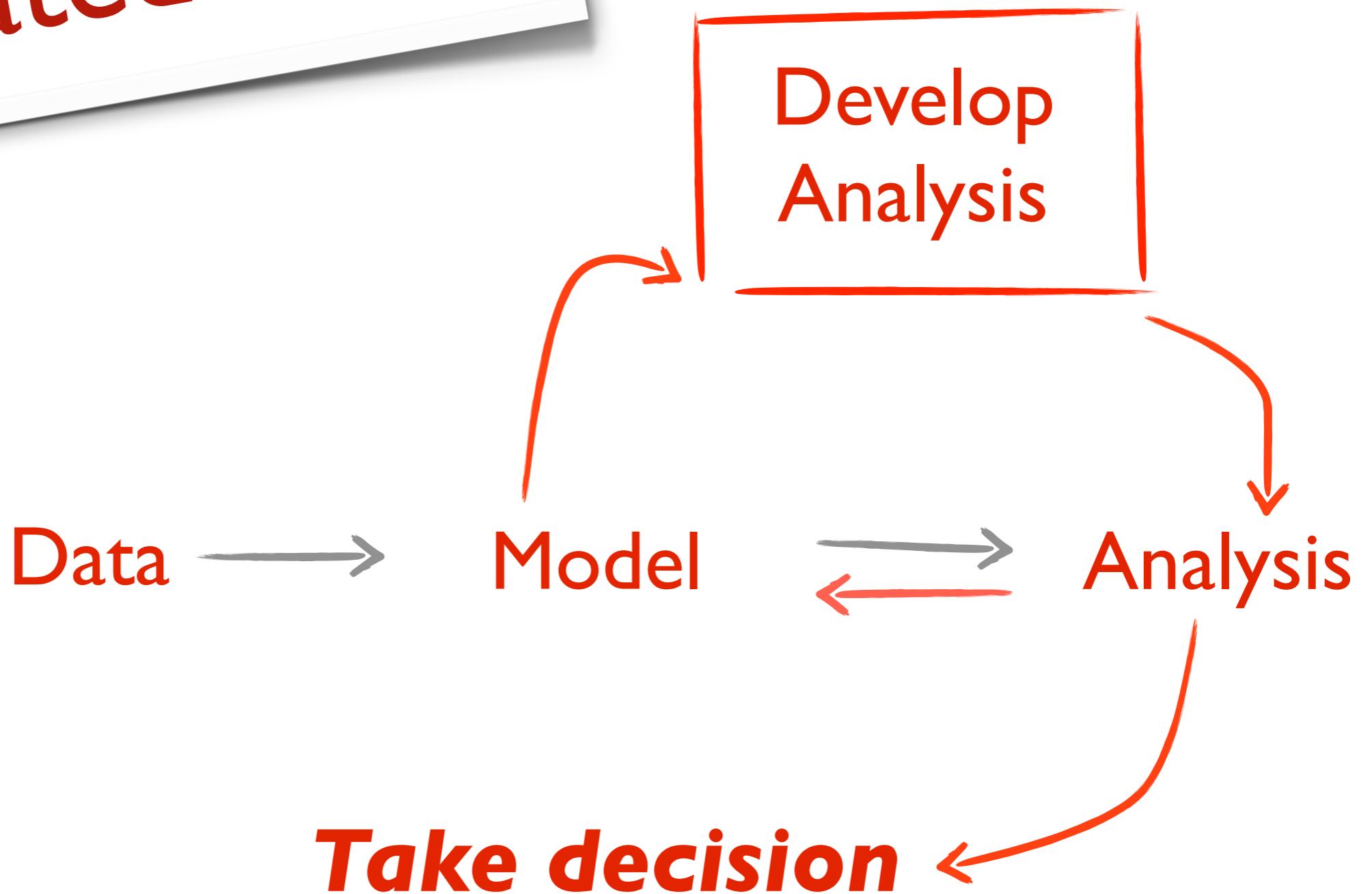
Visualizations



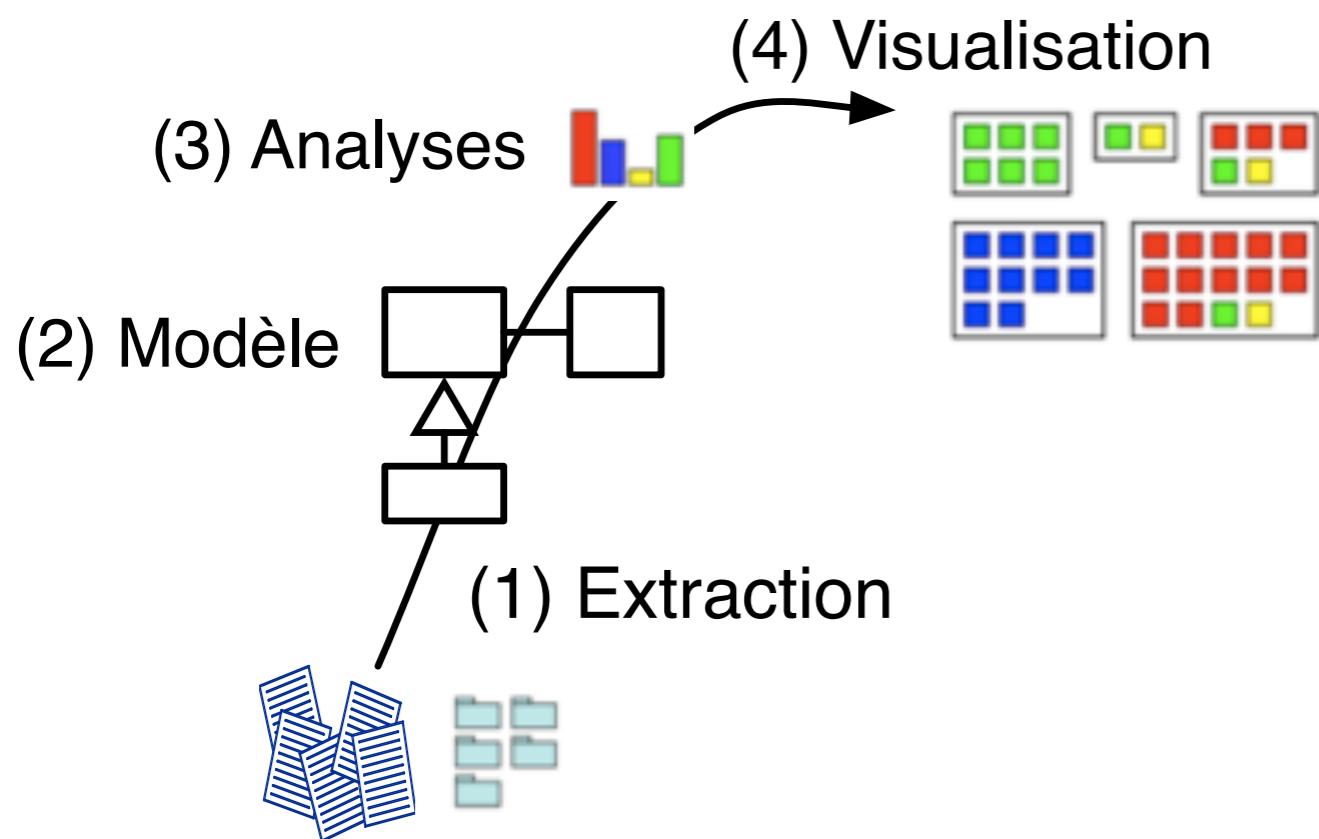
...



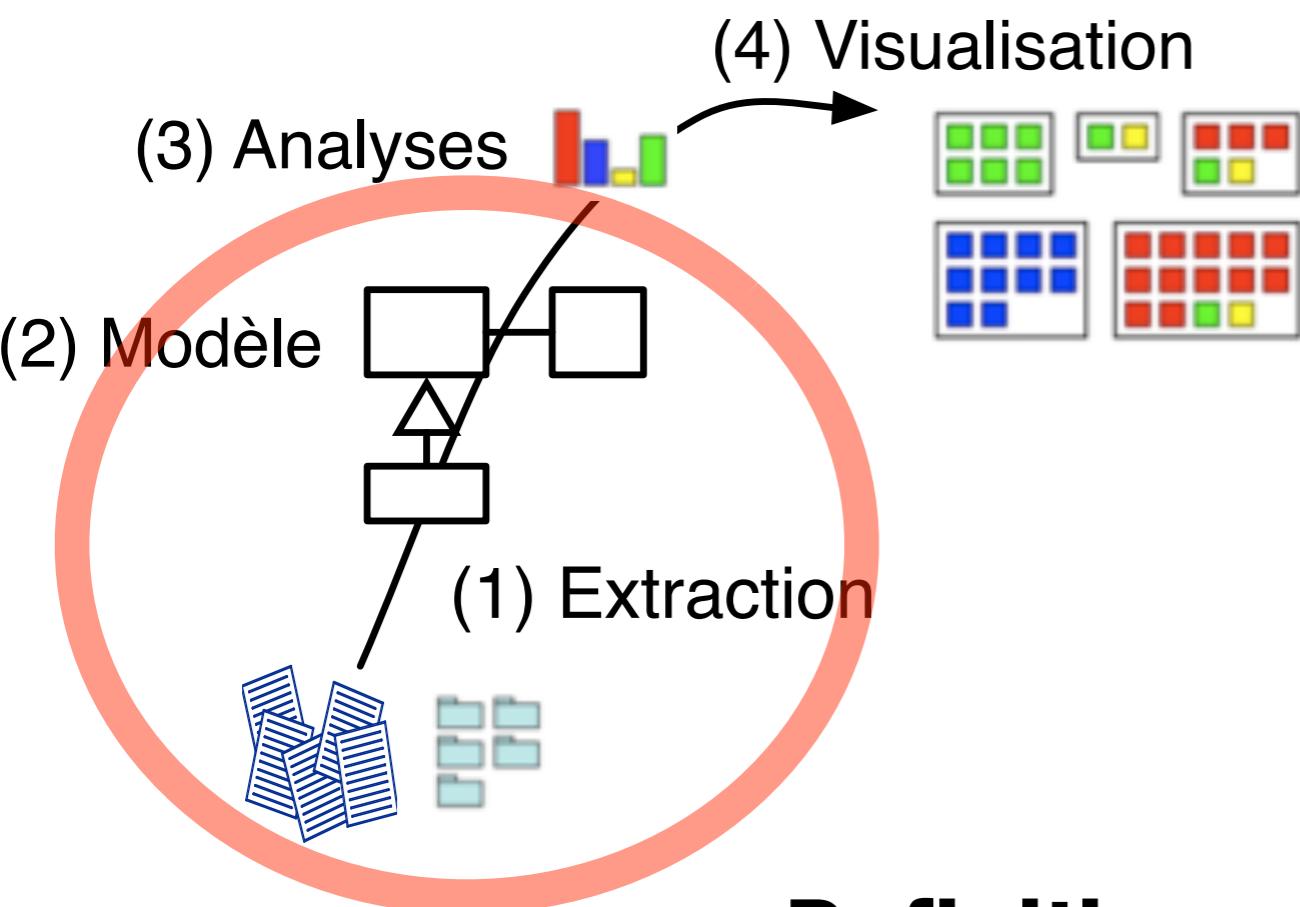
Dedicated Tools



Example : Who is behind package X ?

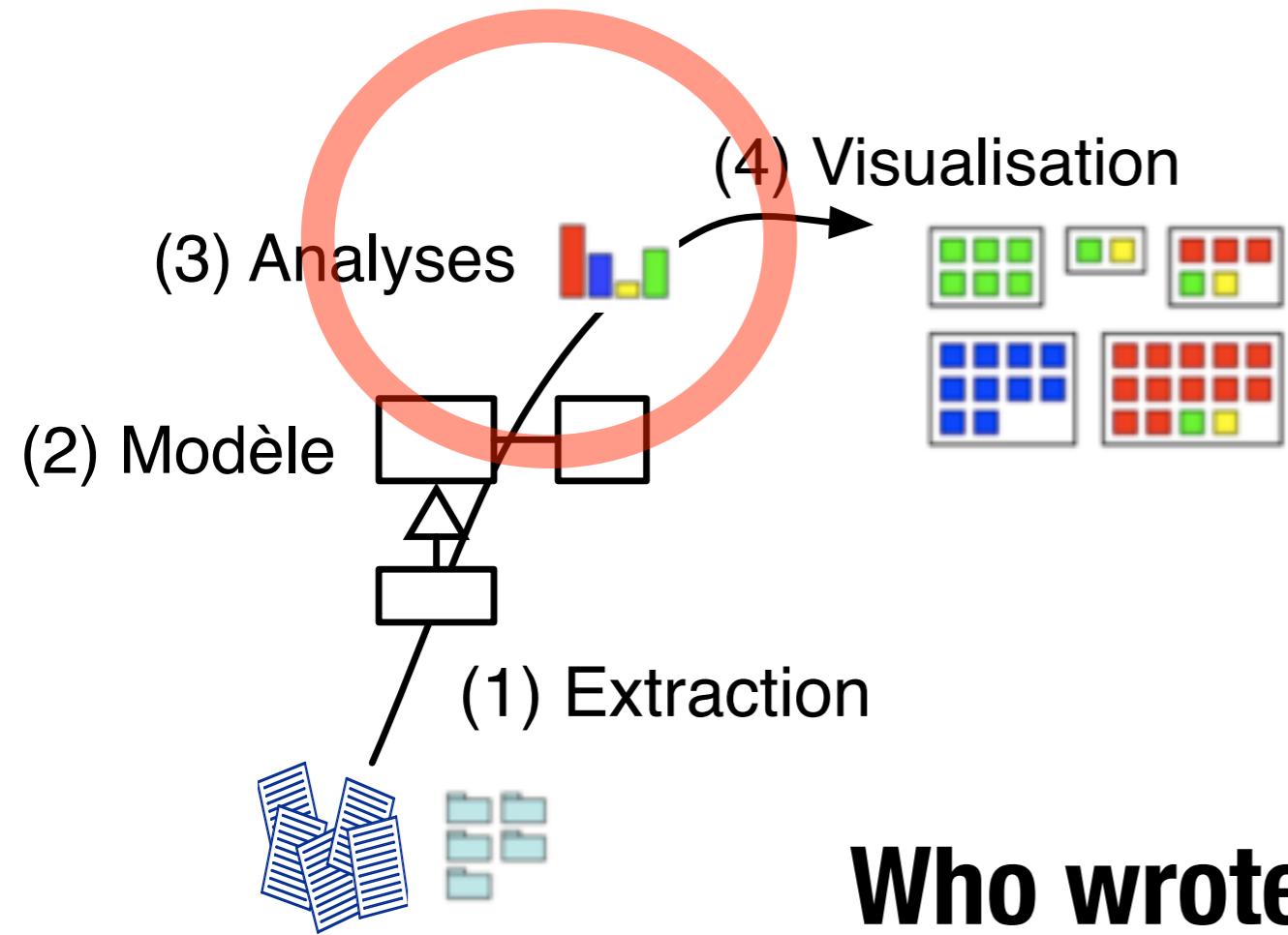


Step 1 - Model Creation/Import



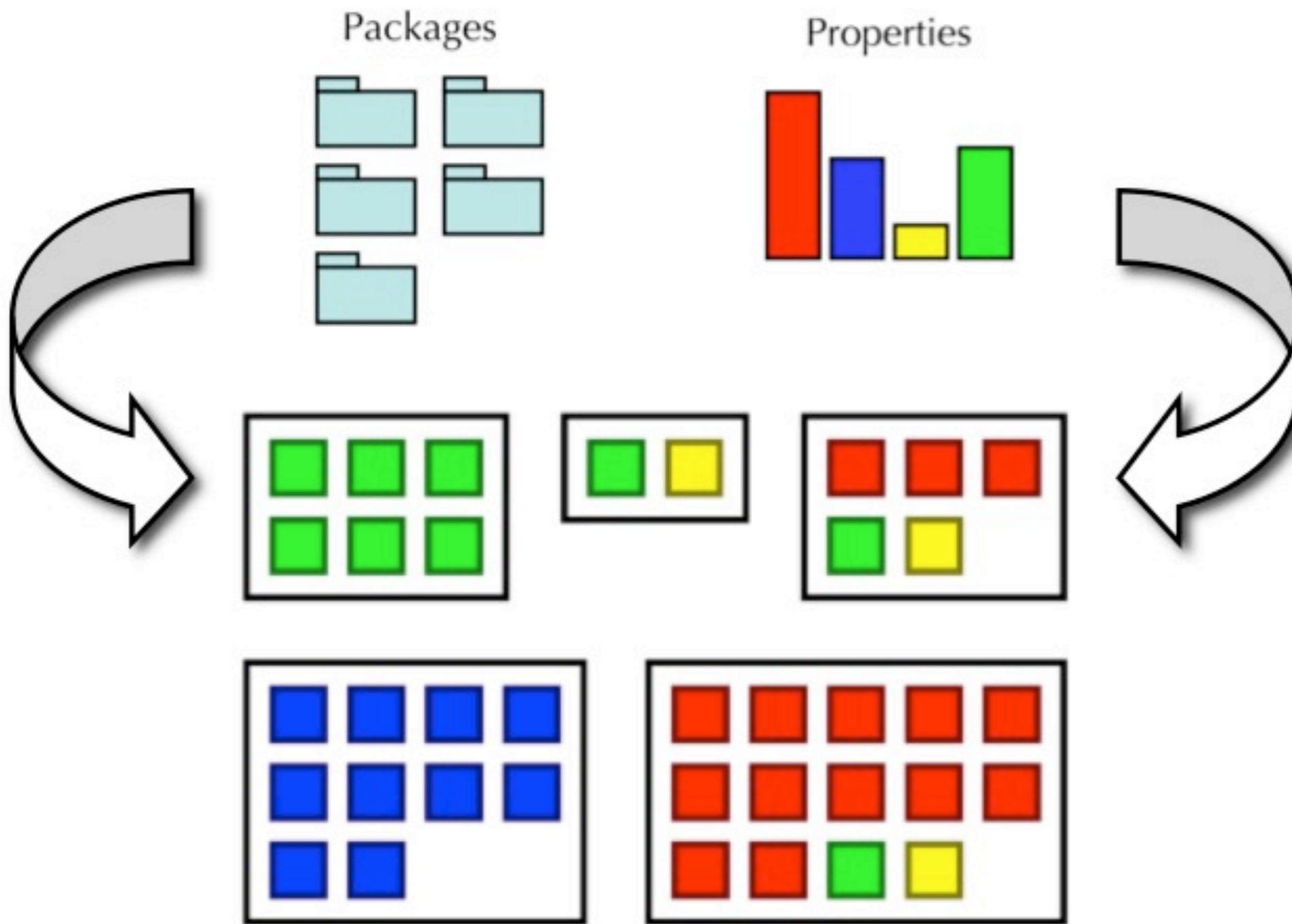
Definition of a model to represent entities
Data Extraction (CSV...)

Step 2 - Analyses

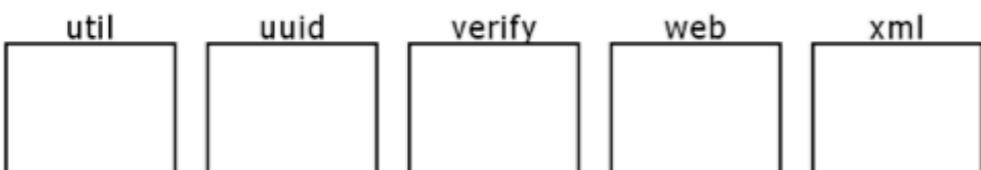
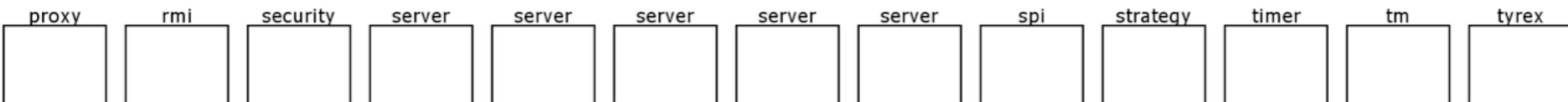
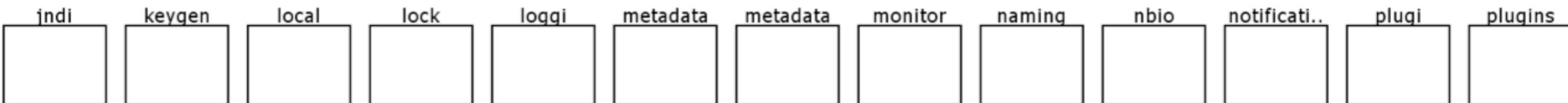
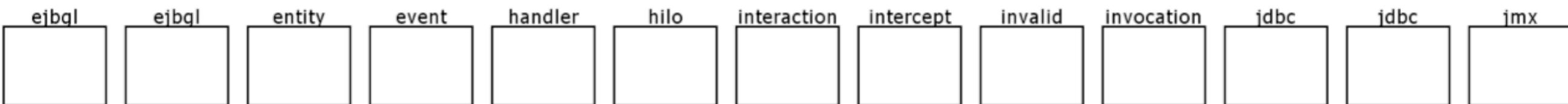
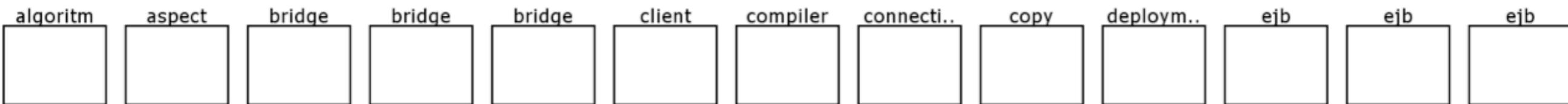


Who wrote how many lines of code?

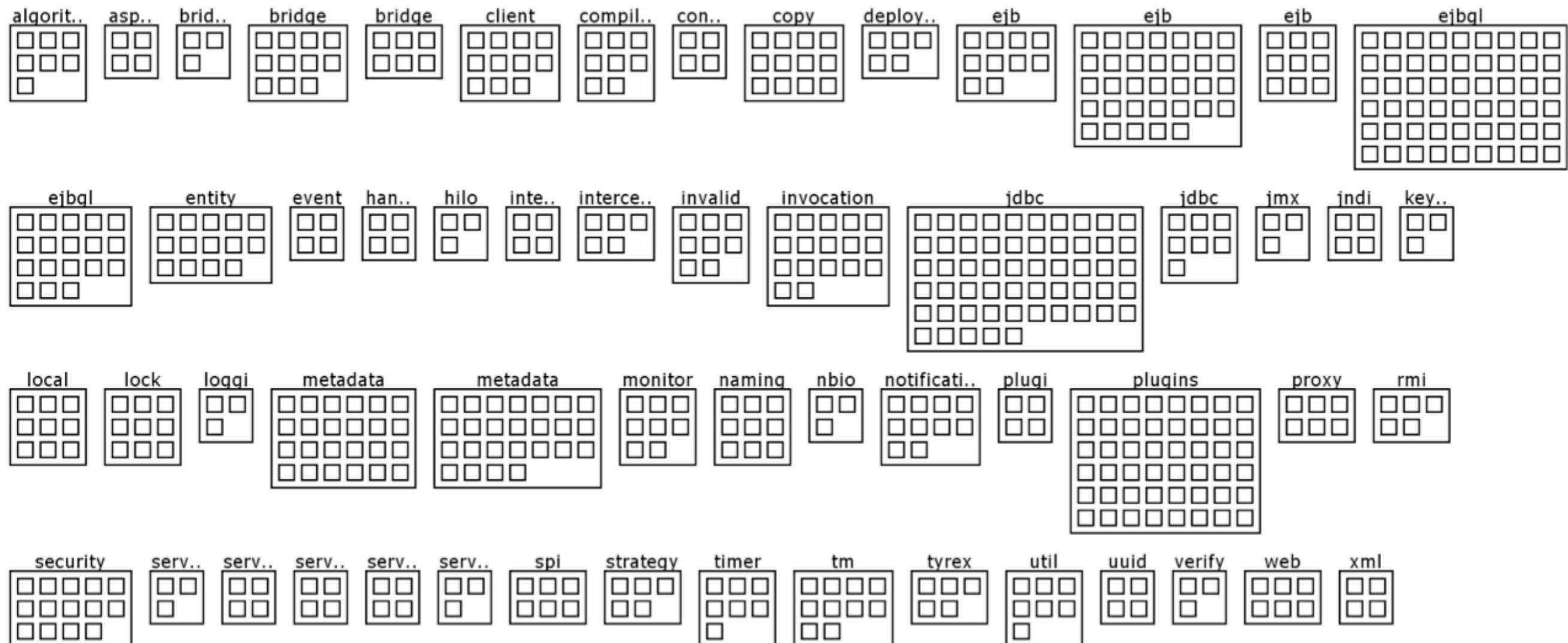
Step : 3 - Creating the Map



Step 1 – for each package draw a rectangle



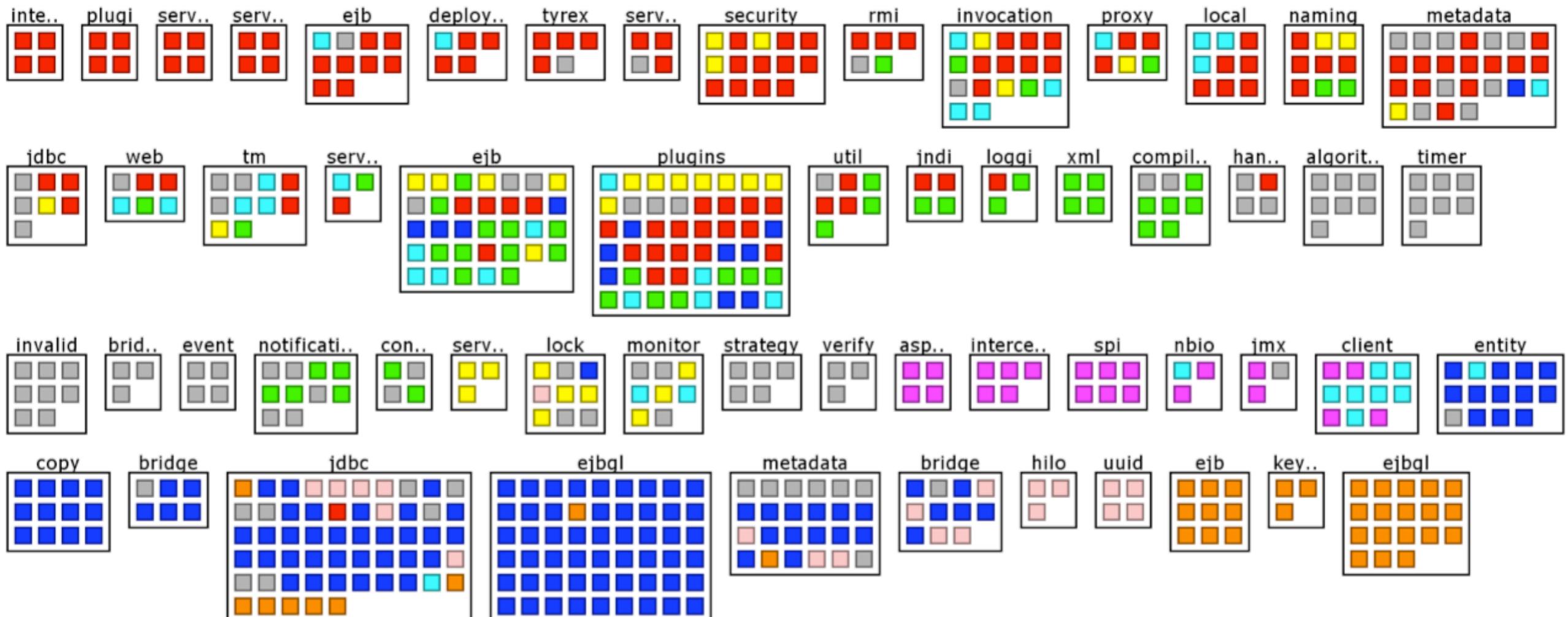
Step 2 – populate packages with classes



Step 3 – color the classes by property

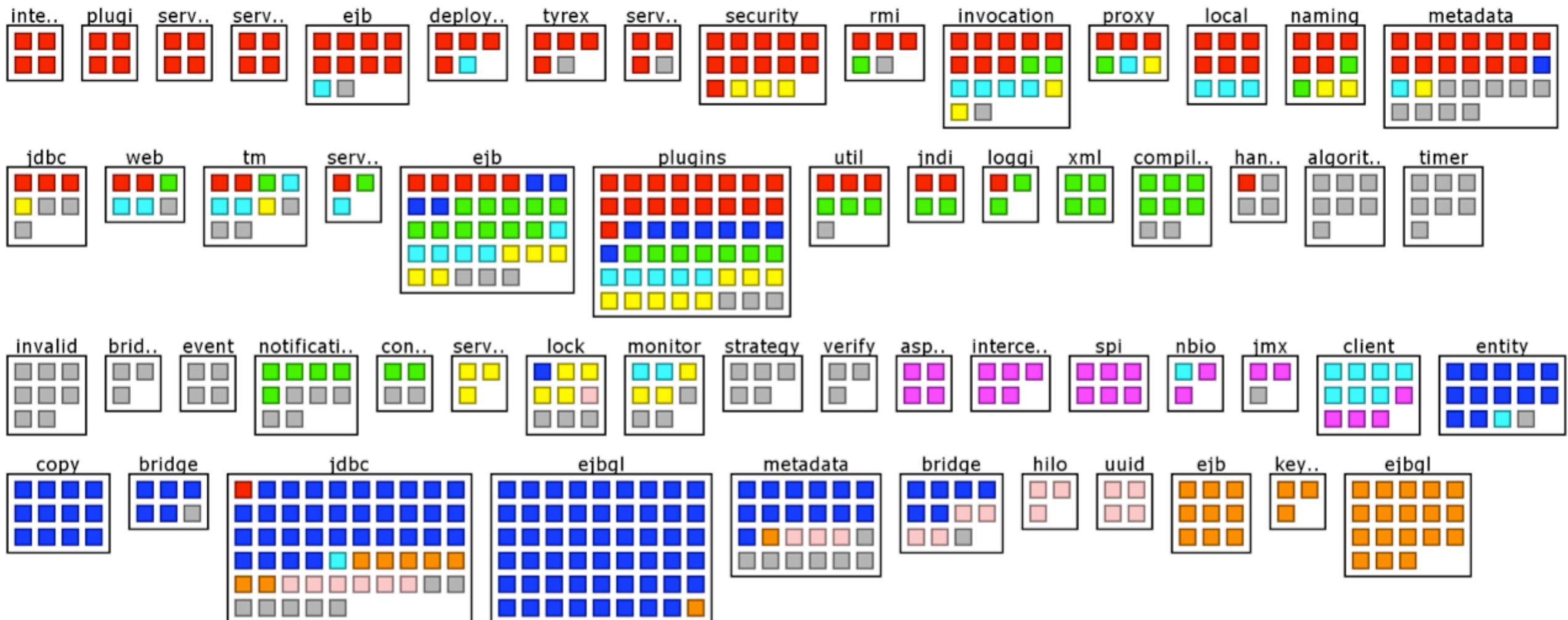


Step 4 – sort packages by content



Sorting with dendrogram seriation.

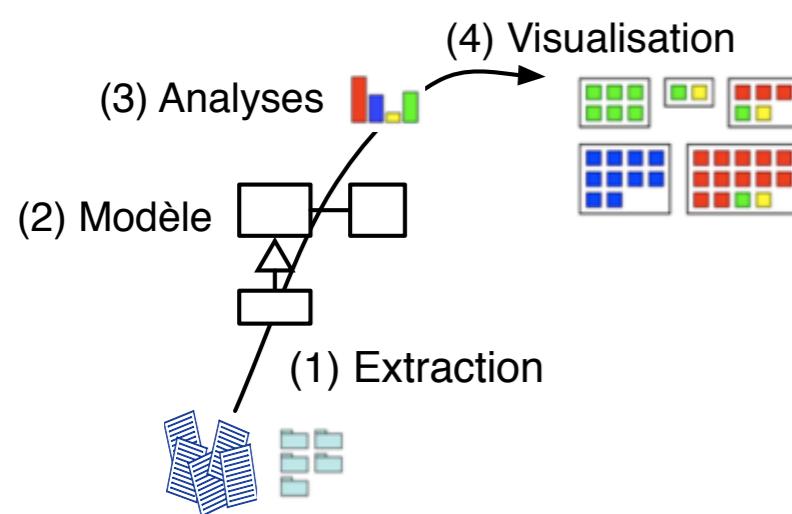
Step 5 – sort classes by properties

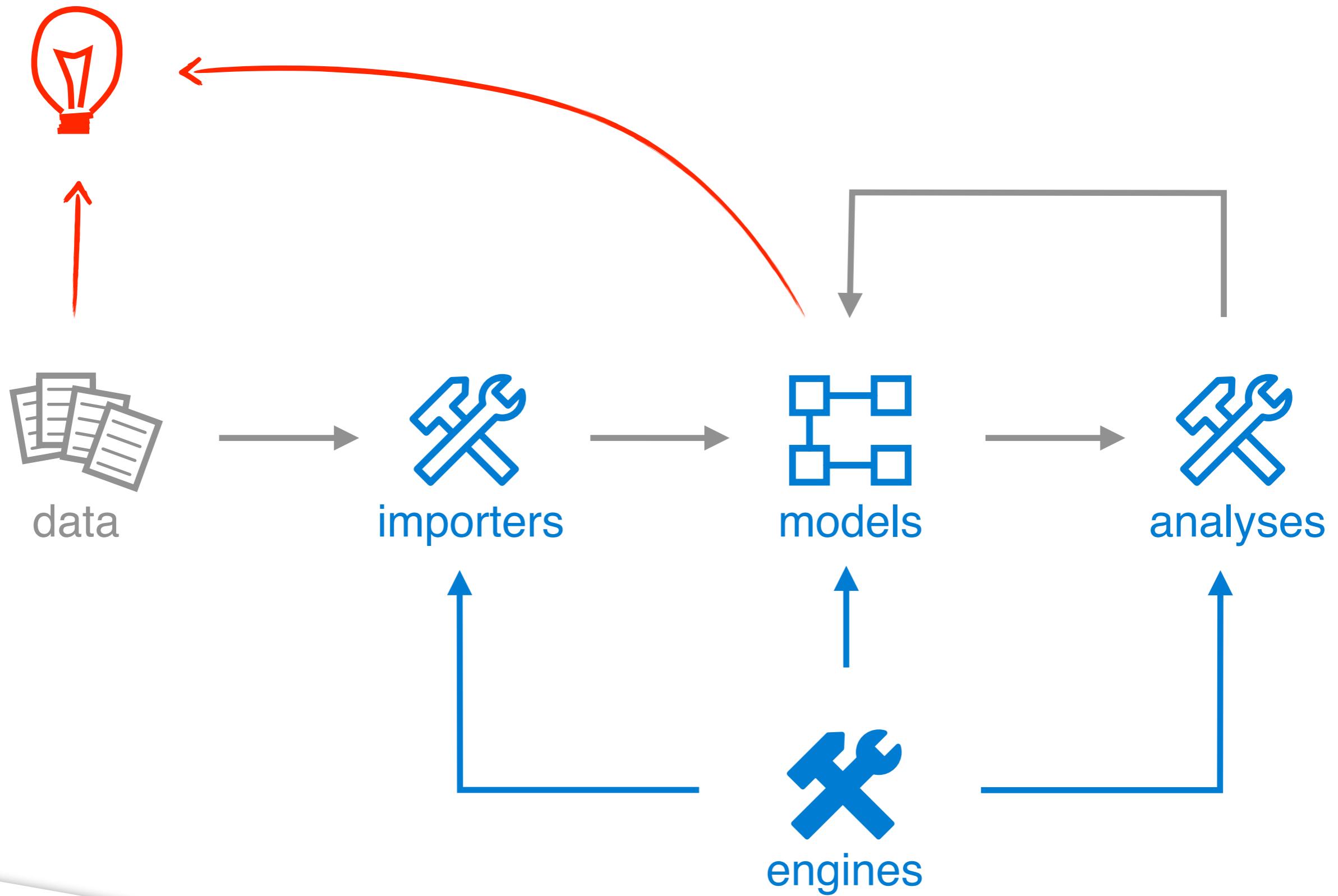




JBoss at a glance

Interactive tool Data in perspective





Navigation in the Model

Moose Panel

Models

- LANExampleJava
- MooseModel
- jedit
- Test

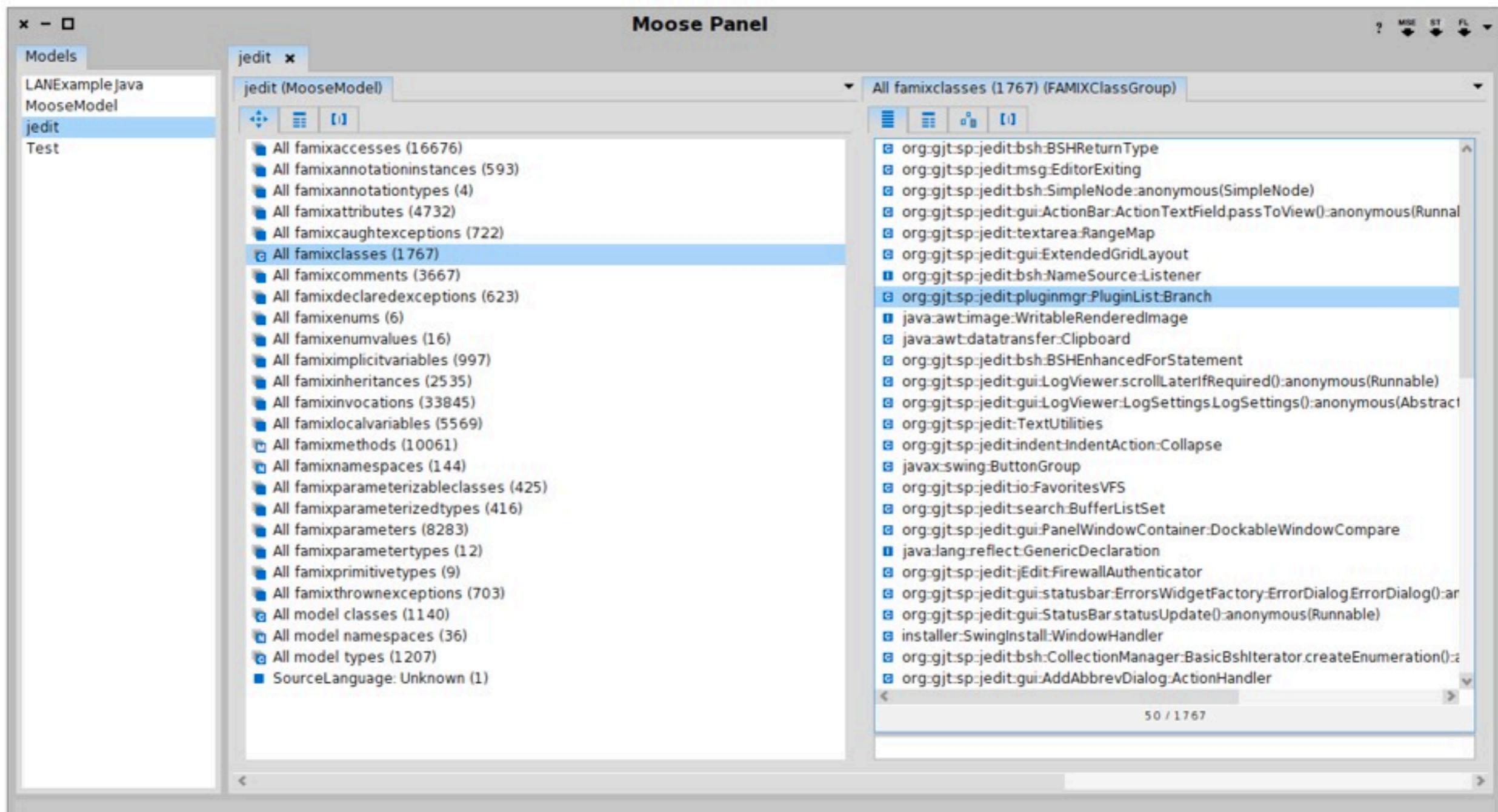
jedit (MooseModel)

- All famixaccesses (16676)
- All famixannotationinstances (593)
- All famixannotationtypes (4)
- All famixattributes (4732)
- All famixcaughtexceptions (722)
- All famixclasses (1767) **(selected)**
- All famixcomments (3667)
- All famixdeclaredexceptions (623)
- All famixenums (6)
- All famixenumvalues (16)
- All famiximplicitvariables (997)
- All famixinheritances (2535)
- All famixinvocations (33845)
- All famixlocalvariables (5569)
- All famixmethods (10061)
- All famixnamespaces (144)
- All famixparameterizableclasses (425)
- All famixparameterizedtypes (416)
- All famixparameters (8283)
- All famixparametertypes (12)
- All famixprimitivetypes (9)
- All famixthrownexceptions (703)
- All model classes (1140)
- All model namespaces (36)
- All model types (1207)
- SourceLanguage: Unknown (1)

All famixclasses (1767) (FAMIXClassGroup)

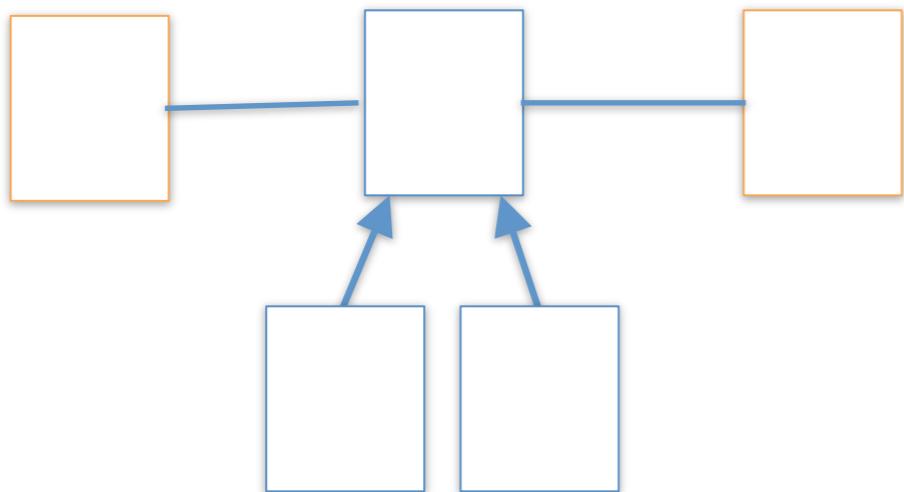
- org:gjt:sp:jedit:bsh:BSHReturnType
- org:gjt:sp:jedit:msg:EditorExiting
- org:gjt:sp:jedit:bsh:SimpleNode:anonymous(SimpleNode)
- org:gjt:sp:jedit:gui:ActionBar:ActionBarTextField.passToView():anonymous(Runnable)
- org:gjt:sp:jedit:textarea:RangeMap
- org:gjt:sp:jedit:gui:ExtendedGridLayout
- org:gjt:sp:jedit:bsh:NameSource:Listener
- org:gjt:sp:jedit:pluginmgr:PluginList:Branch
- java.awt.image:WritableRenderedImage
- java.awt.datatransfer:Clipboard
- org:gjt:sp:jedit:bsh:BSHEnhancedForStatement
- org:gjt:sp:jedit:gui:LogViewer.scrollLaterIfRequired():anonymous(Runnable)
- org:gjt:sp:jedit:gui:LogViewer:LogSettings.LogSettings():anonymous(Abstract
- org:gjt:sp:jedit:TextUtilities
- org:gjt:sp:jedit:indent:IndentAction:Collapse
- javax.swing:ButtonGroup
- org:gjt:sp:jeditio:FavoritesVFS
- org:gjt:sp:jedit:search:BufferListSet
- org:gjt:sp:jedit:gui:PanelWindowContainer:DockableWindowCompare
- java.lang.reflect:GenericDeclaration
- org:gjt:sp:jedit:jEdit:FirewallAuthenticator
- org:gjt:sp:jedit:gui:ErrorsWidgetFactory:ErrorDialog.ErrorDialog():ar
- org:gjt:sp:jedit:gui:StatusBar.statusUpdate():anonymous(Runnable)
- installer:SwingInstall:WindowHandler
- org:gjt:sp:jedit:bsh:CollectionManager:BasicBshIterator.createEnumeration():
- org:gjt:sp:jedit:gui:AddAbbrevDialog.ActionHandler

50 / 1767

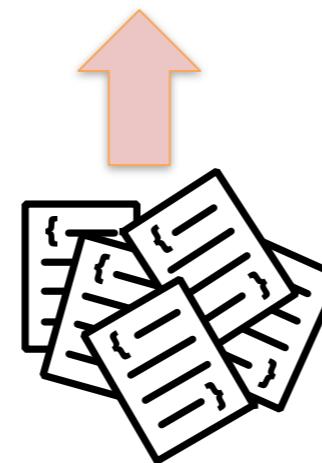
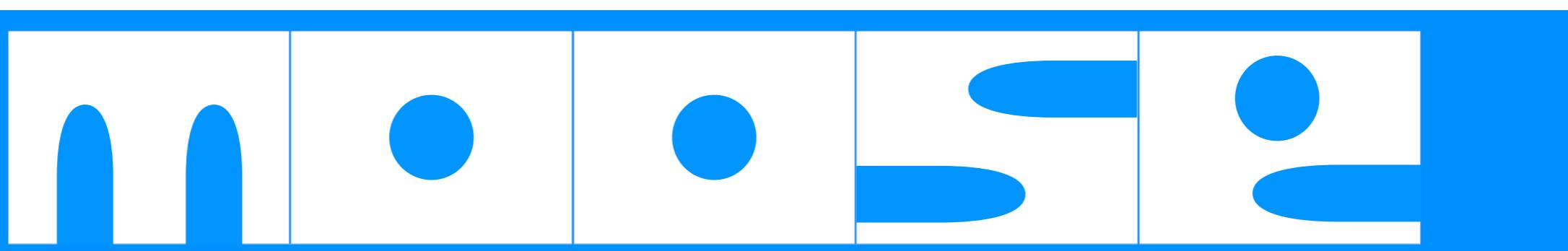




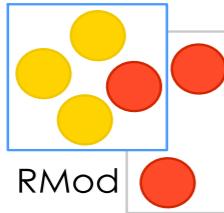
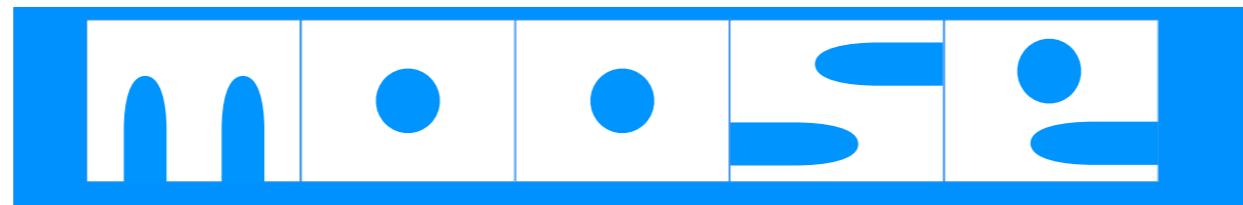
Queries reduce the analysis



Query



Software Metrics



INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

 **INRIA**

centre de recherche LILLE - NORD EUROPE

Metrics in Moose

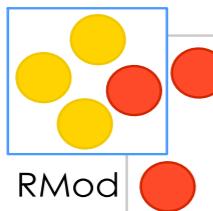
Moose Panel

Models LANExample.java MooseModel jedit Test

All famixclasses (1767) (FAMIXClassGroup)

Properties

Properties	Value
fanIn	0
fanOut	16
hierarchyNestingLevel	1
nameLength	18
numberOfAbstractMethods	0
numberOfAccessesToForeignData	-1
numberOfAccessorMethods	0
numberOfAttributes	4
numberOfAttributesInherited	0
numberOfChildren	0
numberOfComments	18
numberOfConstructorMethods	0
numberOfDuplicatedLinesOfCodeInternally	-1
numberOfExternalDuplications	0
numberOfInternalDuplications	0
numberOfLinesOfCode	1026
numberOfMessageSends	-17
numberOfMethodProtocols	0
numberOfMethods	17
numberOfMethodsAdded	16
numberOfMethodsInHierarchy	90
numberOfMethodsInherited	73
numberOfMethodsOverridden	1
numberOfParents	2
numberOfPrivateAttributes	4
numberOfPrivateMethods	4
numberOfProtectedAttributes	0
numberOfProtectedMethods	0



Metrics Dashboard

Glamorous Browser

Tests percentage of success
100 %

0 successes
0 failures
0 errors
0 milliseconds to run

Rule compliance
99.84 %

13 violations
⚠ 6 important violations
7 warnings

Duplicated lines
4.0 %

9 duplications
360 duplicated lines

Number of lines of code
7823

27 classes
94 packages
7.58 LOC / method
⚠ 38.22 methods / class
0.28 classes / package

Percentage of methods commented
66 %

2 undocumented classes
350 undocumented methods
92 % classes commented

Number of alerts
3

Number of important violations > 0
Average number of methods per class > 7
Average complexity per class > 40

Top LOC package
Collections-Abstract 4035 LOC (51%)

10% of the packages: 4035 LOC (51%)
50% of the packages: 7493 LOC (95%)

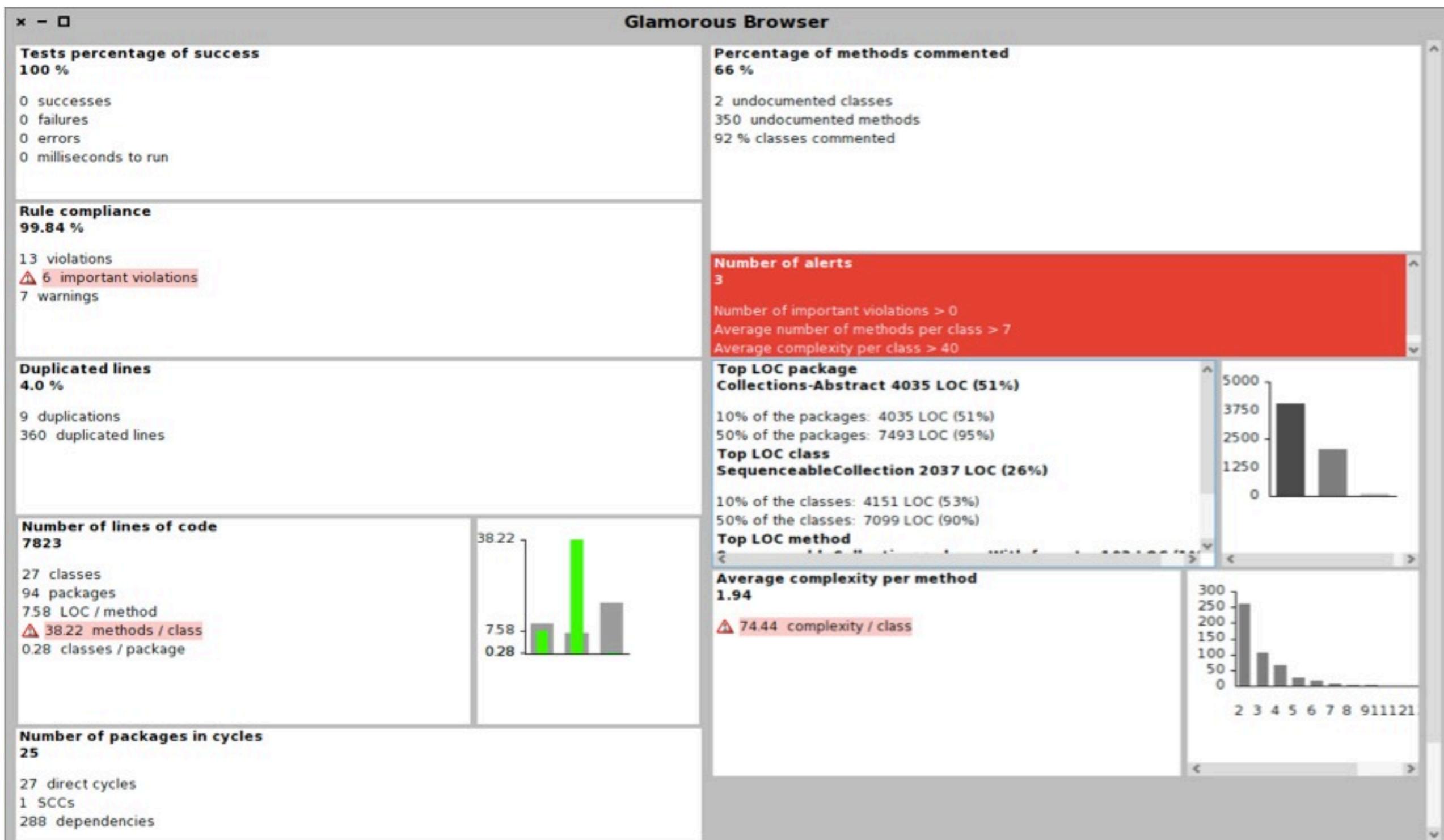
Top LOC class
SequenceableCollection 2037 LOC (26%)

10% of the classes: 4151 LOC (53%)
50% of the classes: 7099 LOC (90%)

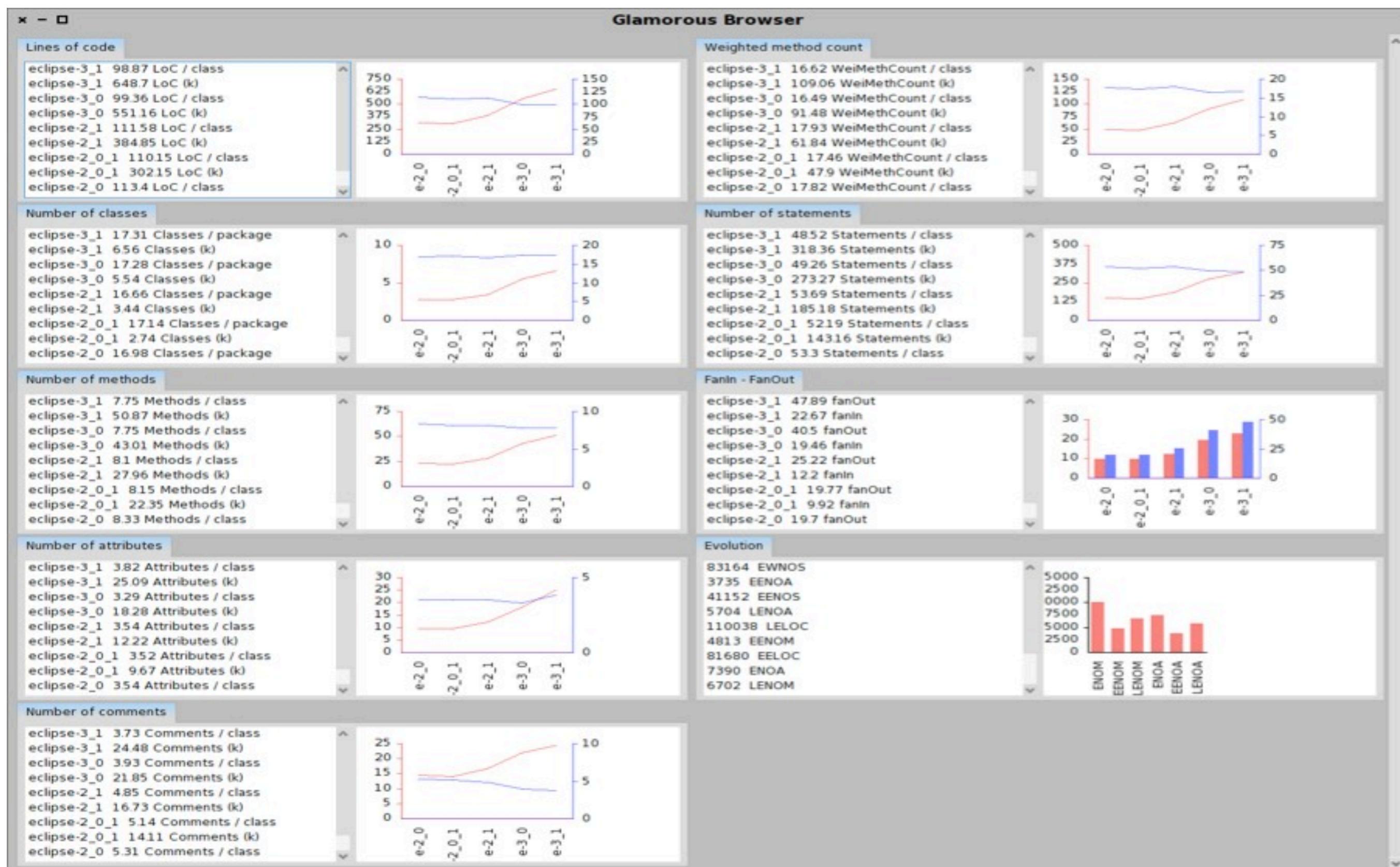
Top LOC method
⚠

Average complexity per method
1.94

⚠ 74.44 complexity / class



Evolution Dashboard

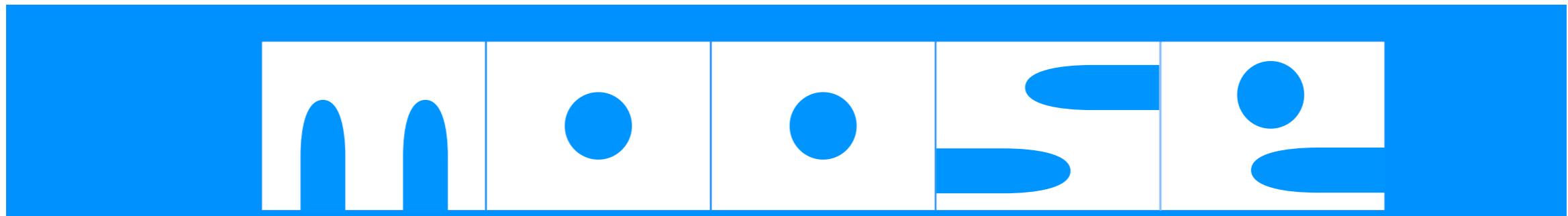
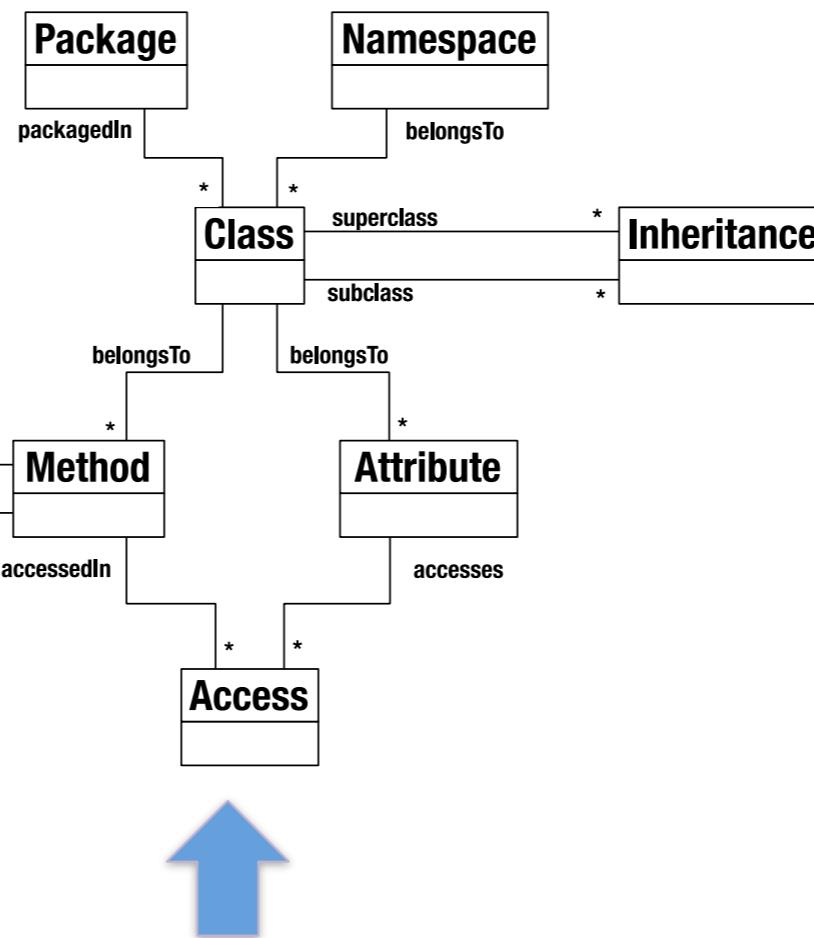


Evolution Matrix

...	org:jhotdraw:draw:BoxHandleKit::SouthWestHandle
.....	java.awt.GraphicsEnvironment
.......	java.awt.EventQueue
....	CH:jfa:draw:standard:PasteCommand
.....	java.awt.datatransfer.Clipboard
....	CH:jfa:draw:figures:ShortestDistanceConnector
.....	org:jhotdraw:geom:Insets2D::Float
.....	javax.swing.plaf.SliderUI
....	org:jhotdraw:draw:TextCreationTool
....	org:jhotdraw:draw:handle:ResizeHandleKit::SouthEastHandle
.....	javax.swing.JOptionPane
....	CH:jfa:draw:figures:ConnectedTextTool::DeleteUndoActivity
.....	org:jhotdraw:draw:DefaultDrawingViewBeanInfo
....	CH:jfa:draw:contrib:html:FigureContentProducerContext
....	CH:jfa:draw:test:figures:ScribbleToolTest
.....	java.lang.InterruptedException
....	org:jhotdraw:application:OSXPaletteHandler
....	org:jhotdraw:test:contrib:JScrollPaneDesktopTest
.....	java.awt.Graphics2D
....	org:jhotdraw:figures:ConnectedTextTool::DeleteUndoActivity
.....	java.awt.event.ItemEvent

Moose Meta-Model

To represent program information



Java

Smalltalk

C/C++

.NET

...

Custom rules can be captured and checked
daily for you.

Dedicated Analysis reports with Moose provide pertinent analysis for software elements

Report

Remote methods called without error handling

Index

Server

- No @Interceptors in
- Wrong @Interceptor
- Remote methods ca
- Usages of EntityMan
- Direct EntityManager
- Entity Beans not in p
- Entities that are Ser
- Embeddeable not Se
- Not allowed calls to
- Serializable wihtout
- Serializable inner cla

Server API

- TOs not serializable
- TOs with non private
- TOs with only static
- TOs with qualified p
- TOs not used

All calls to remote methods should go through an implementation of

41 remote methods

app:browser:srv:api:IFavorites

app:browser:srv:api:IFavorites

app:browser:srv:api:IHistorySe

app:browser:srv:api:IHistorySe

common:legacyentities:srv:api

common:legacyentities:srv:api

common:legacyentities:srv:api

common:legacyentities:srv:api

core:login:srv:api:ILoginService

core:login:srv:api:ILoginService

core:login:srv:api:ILoginService

core:login:srv:api:ILoginService

core:login:srv:api:ILoginService

core:login:srv:api:ILoginService

core:login:srv:api:ILoginService

core:login:srv:api:ILoginService

core:programmenu:srv:api:Pro

srv:common:service:api:Prop

srv:common:service:api:Serv

srv:masterdata:api:IMasterDat

srv:masterdata:api:IMasterDat

srv:masterdata:api:IMasterDat

srv:masterdata:api:IMasterDat

srv:masterdata:api:IMasterDat

srv:masterdata:api:IMasterDat

srv:masterdata:api:IMasterDat

srv:masterdata:api:IMasterDat

srv:masterdata:api:IMasterDat

masterdata:api:IMasterDataService.getHierarchy()

getHierarchy

run

getChildElements

getVersionisedMDHNode

getLEPNode

lepToLerf

getHierarchy

getCatalogItemTO

errorhandledInitialize

getElements

getChildElement

getChildren

getChildElement

getDuration—derivePrescriptionDTO—

createLebensbereichCategory—createLe

completeSessionEntity—saveSess

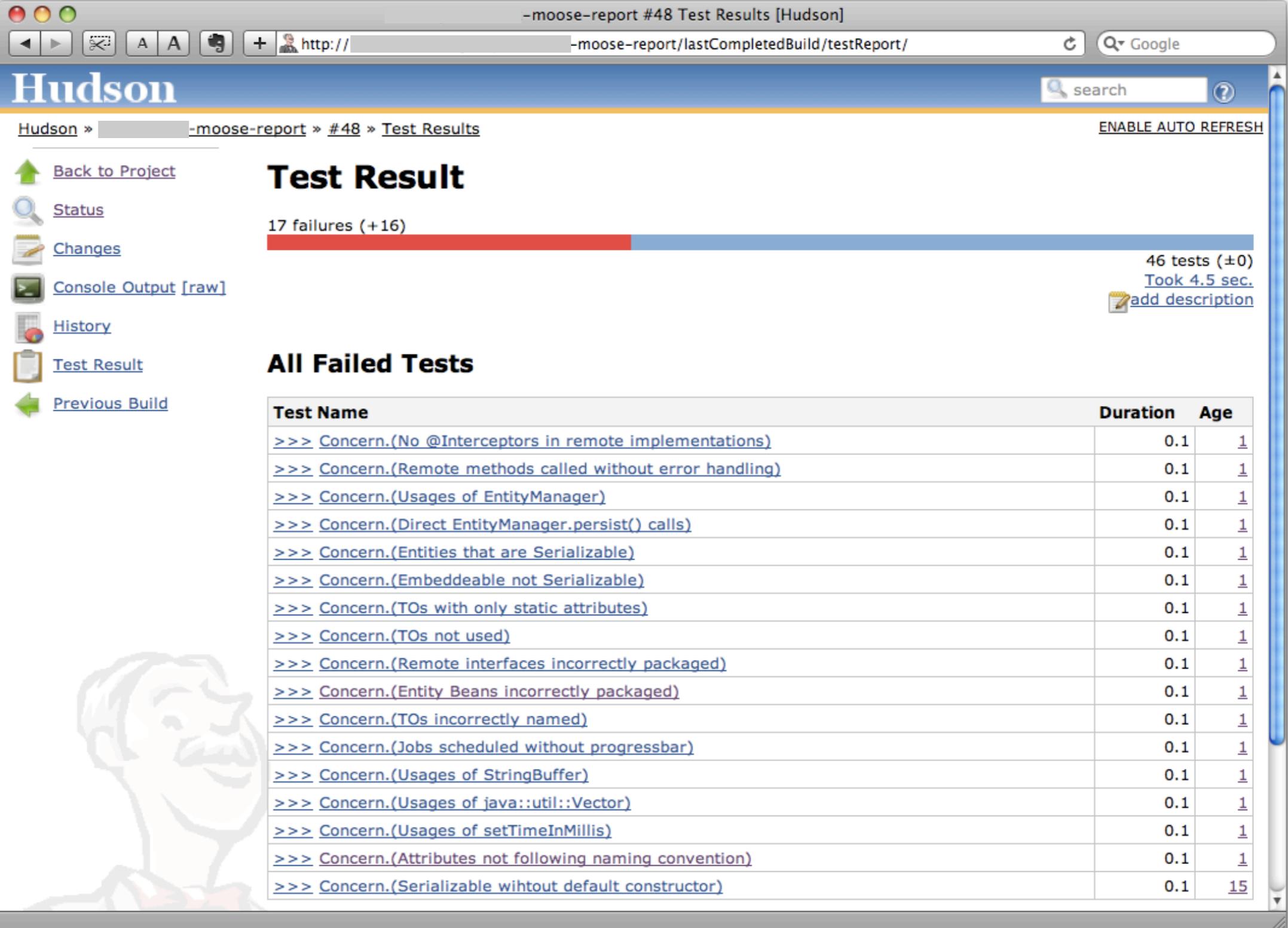
getSessions

resolveCatalogItemHierarchyForEntr

validateCase

```
/**  
 * Provides a hierarchy, mainly used for trees.  
 *  
 * @param query  
 *      - identifies the catalog  
 * @return A tree of nodes matching the query.  
 */  
MDHNode getHierarchy(MDItemHierarchyQueryTO query);
```

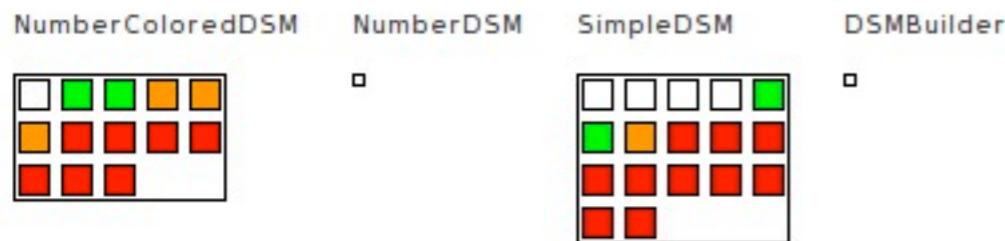
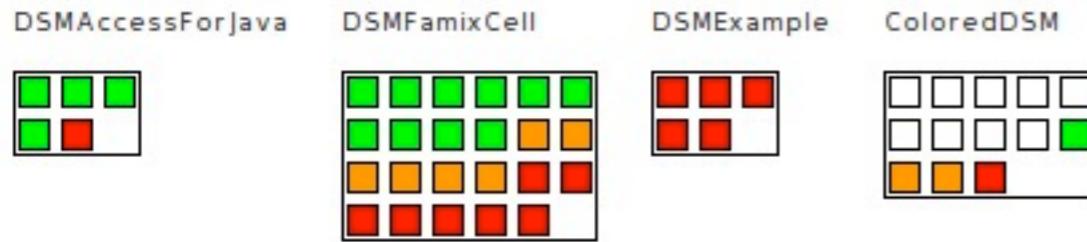
Reports can be integrated into software process: Automated, continuous reports with integration servers



The screenshot shows a Hudson test results page for a build named '-moose-report #48'. The main title is 'Test Result' with a subtitle 'All Failed Tests'. A summary bar indicates 17 failures (+16) in red and 46 tests (±0) in blue. The Hudson logo is visible on the left.

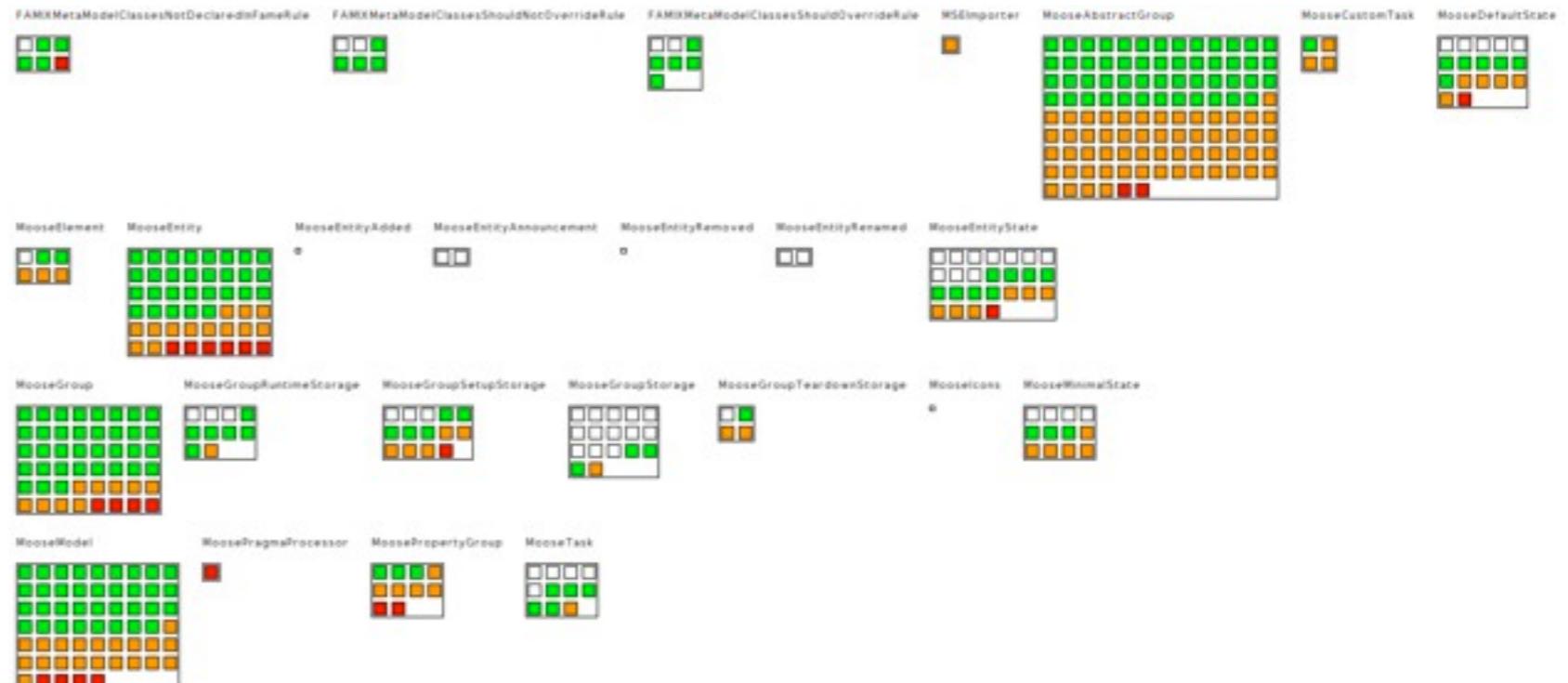
Test Name	Duration	Age
>>> Concern.(No @Interceptors in remote implementations)	0.1	1
>>> Concern.(Remote methods called without error handling)	0.1	1
>>> Concern.(Usages of EntityManager)	0.1	1
>>> Concern.(Direct EntityManager.persist() calls)	0.1	1
>>> Concern.(Entities that are Serializable)	0.1	1
>>> Concern.(Embeddable not Serializable)	0.1	1
>>> Concern.(TOs with only static attributes)	0.1	1
>>> Concern.(TOs not used)	0.1	1
>>> Concern.(Remote interfaces incorrectly packaged)	0.1	1
>>> Concern.(Entity Beans incorrectly packaged)	0.1	1
>>> Concern.(TOs incorrectly named)	0.1	1
>>> Concern.(Jobs scheduled without progressbar)	0.1	1
>>> Concern.(Usages of StringBuffer)	0.1	1
>>> Concern.(Usages of java::util::Vector)	0.1	1
>>> Concern.(Usages of setTimeInMillis)	0.1	1
>>> Concern.(Attributes not following naming convention)	0.1	1
>>> Concern.(Serializable without default constructor)	0.1	15

Custom Rules

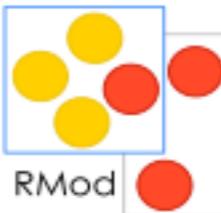
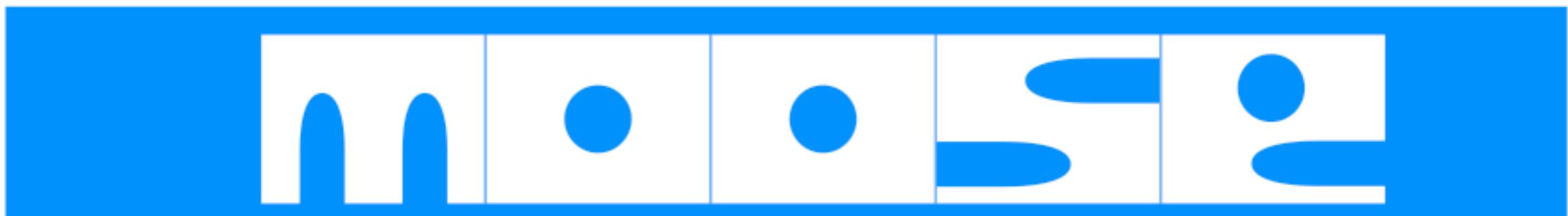
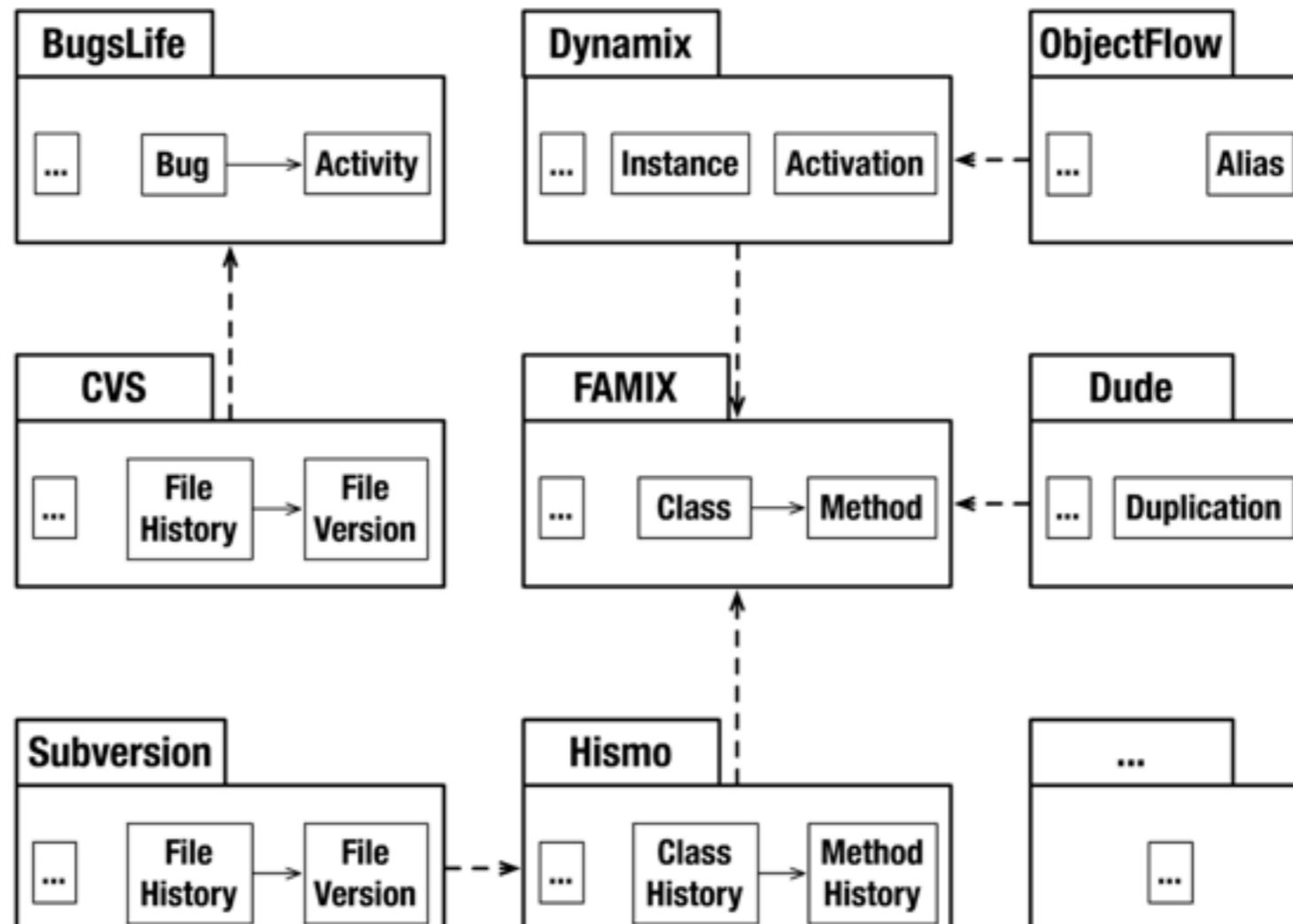


Rule followers and breakers

green: followers
orange: break < 2 rules
Red: break > 2 rules



Extensible Meta model

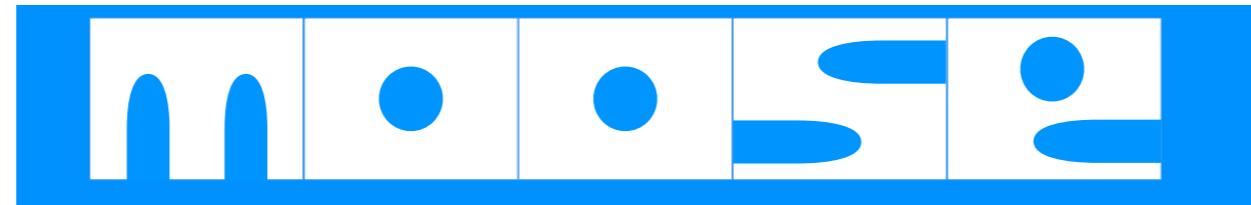


INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

 **INRIA**

centre de recherche LILLE - NORD EUROPE

Tools Design With Moose



Modeling of Eclipse Plugins

The screenshot shows a modeling interface with two main panes. The left pane is titled "Papyrus-Plugins (MooseModel)" and contains a list of items:

- All famixedeclipsepluginimports (1610)
- All famixedeclipseplugins (306)
- SourceLanguage: Unknown (1)

The right pane is titled "All famixedeclipseplugins (306) (FAMIXEclipsePluginGroup)" and also contains a list of items:

- org.apache.xml.resolver
- org.eclipse.papyrus.widgets.gmf
- org.eclipse.papyrus.diagram.usecase
- org.eclipse.papyrus.table.menu
- org.eclipse.gmf.runtime.diagram.ui.printing

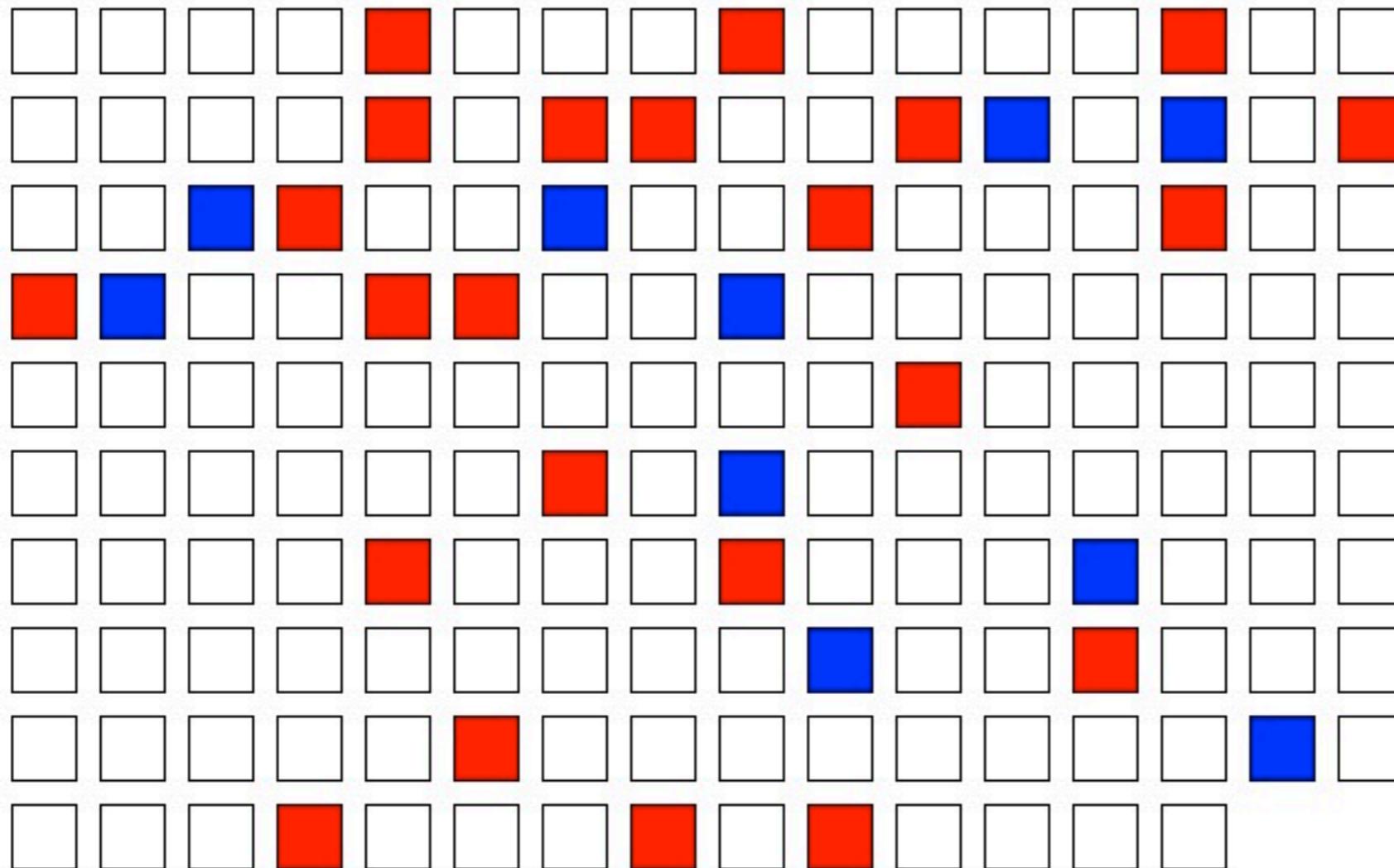
The screenshot shows a modeling interface with two main panes. The left pane is titled "org.eclipse.gm...m.ui.printing (FAMIXEclipsePlugin)" and contains a list of items:

- IncomingFAMIXEclipsePluginImport: org.eclipse.papyrus.diagram.common r

The right pane is titled "org.eclipse.pa...m.ui.printing (FAMIXEclipsePluginImport)" and also contains a list of items:

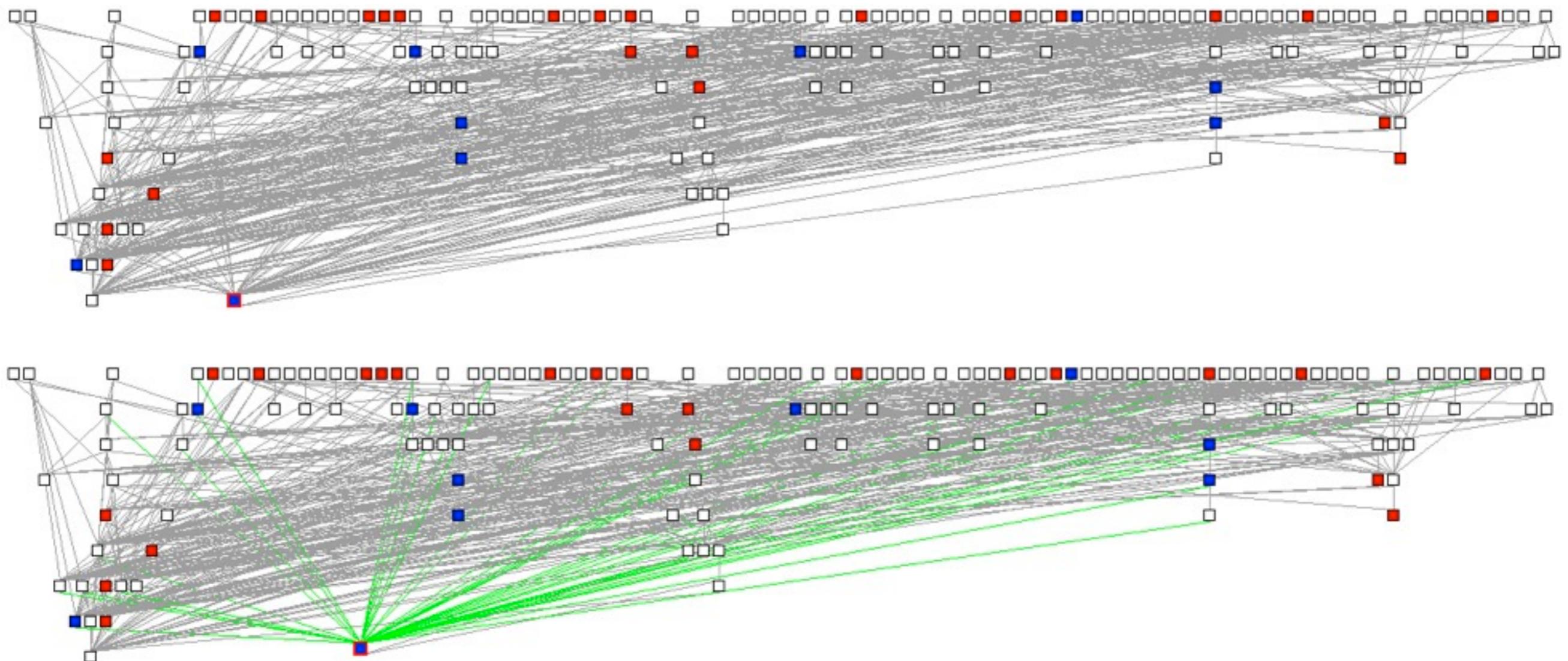
- From: org.eclipse.papyrus.diagram.common (1)
- Source: org.eclipse.papyrus.diagram.common (1)
- Target: org.eclipse.gmf.runtime.diagram.ui.printing (1)
- To: org.eclipse.gmf.runtime.diagram.ui.printing (1)

Contextual Visualization of Plugins



Core: Blue
Uml: Red

Visualization of Plugins & Dependencies



All incoming dependencies for a package in green

Analysis of bugs

BugMaps

org:eclipse:jdt:core:dom:MethodBinding

```

/*
public boolean isConstructor() {
    return this.binding.isConstructor();
}

/*
 * @see IMethodBinding#isDefaultConstructor()
 * @since 3.0
 */
public boolean isDefaultConstructor() {
    if (this.binding.declaringClass.isBinaryBinding()) {
        return false;
    }
}

```

BugMaps

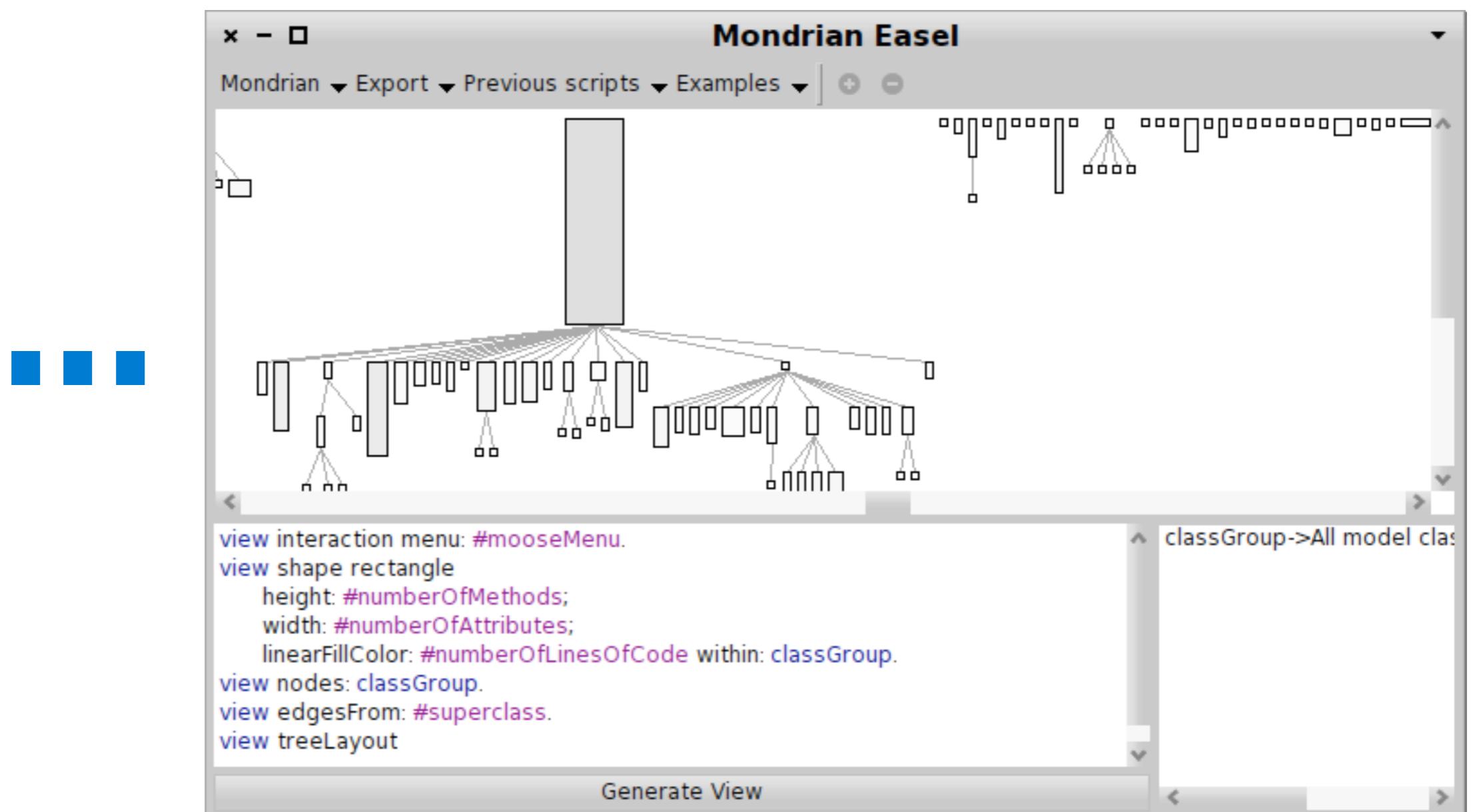
Metric

Metric	Value
evolutionOfNumberOfClasses	7
stable	false
annotationInstances	#0
balanced	false
age	11

Evolution of NOB

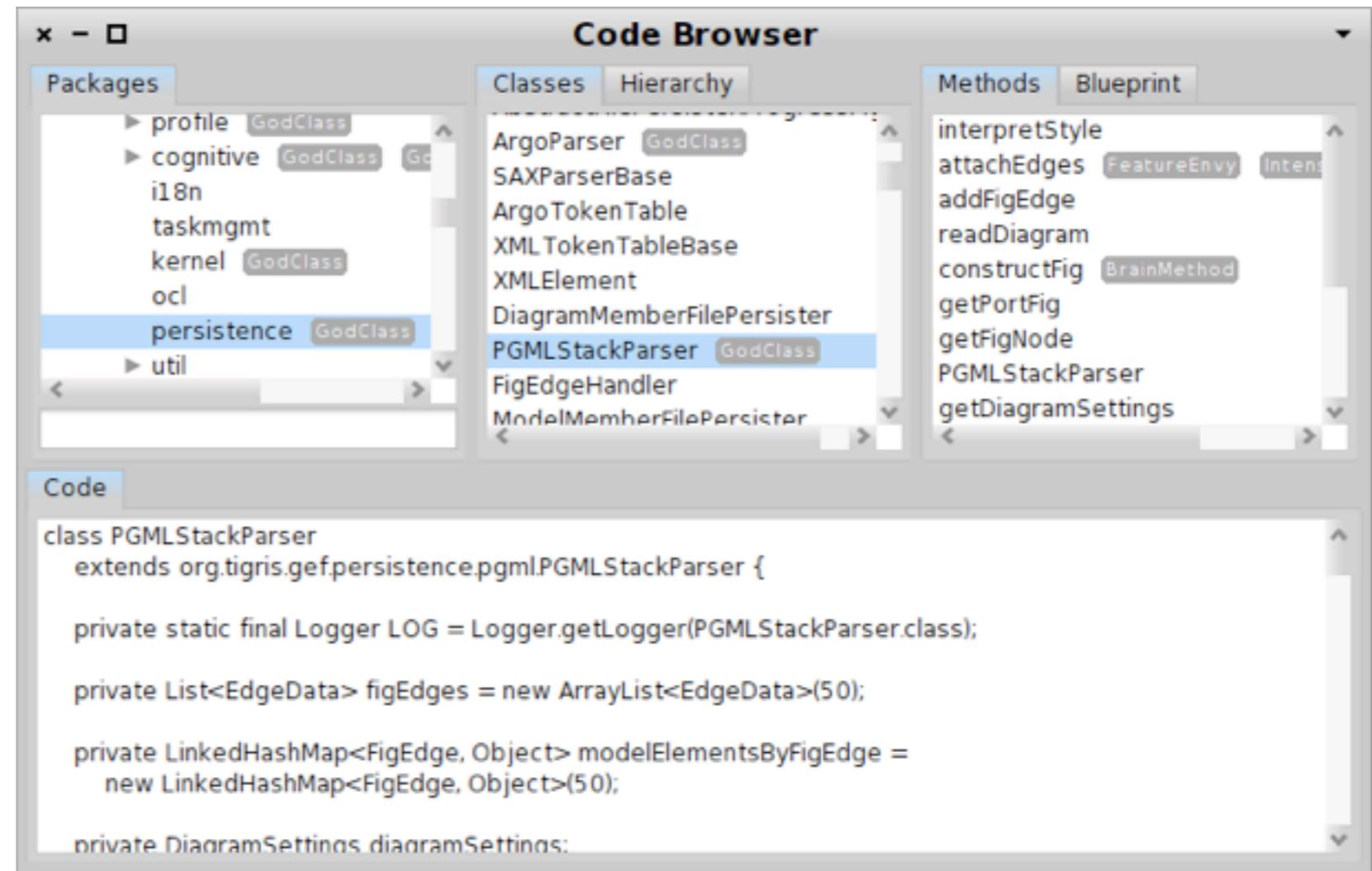
Classes with bugs

Dedicated Analyses



Dedicated Tools

Rich
Compact
Best Focus



The screenshot shows a 'Code Browser' window with three main panes: Packages, Classes, and Methods.

- Packages:** Shows a tree view of packages: profile, cognitive, i18n, taskmgmt, kernel, ocl, persistence (selected), and util.
- Classes:** Shows a list of classes: ArgoParser, SAXParserBase, ArgoTokenTable, XMLTokenTableBase, XMLElement, DiagramMemberFilePersister, PGMLStackParser (selected), FigEdgeHandler, and ModelMemberFilePersister.
- Methods:** Shows a list of methods: interpretStyle, attachEdges, addFigEdge, readDiagram, constructFig, getPortFig, getFigNode, PGMLStackParser, and getDiagramSettings.

Code: A code editor pane showing the source code for the PGMLStackParser class:

```
class PGMLStackParser
    extends org.tigris.gef.persistence.pgml.PGMLStackParser {

    private static final Logger LOG = Logger.getLogger(PGMLStackParser.class);

    private List<EdgeData> figEdges = new ArrayList<EdgeData>(50);

    private LinkedHashMap<FigEdge, Object> modelElementsByFigEdge =
        new LinkedHashMap<FigEdge, Object>(50);

    private DiagramSettings diagramSettings;
```

```
b := GLMTabulator new.  
b column: #namespaces;  
column: #classes;  
column: #methods.  
b transmit to: #namespaces;  
andShow: [:a |  
    a tree  
    display: [ :model |  
        model allNamespaces  
        select: #isRoot ];  
    children: #childScopes;  
    format: #name ].  
b transmit to: #classes;  
from: #namespaces;  
andShow: [:a |  
    a list  
    display: #classes;  
    format: #name ].  
b transmit to: #methods;  
from: #classes;  
andShow: [:a |  
    a list  
    display: #methods;  
    format: #signature ].
```

```
b transmit  
    toOutsidePort: #class;  
    from: #classes.  
b transmit to: #methods;  
    from: #methods.  
B := GLMTabulator new.  
B title: 'Code Browser'.  
B row: #nav;  
    row: #details.  
B transmit to: #nav;  
andShow: [:a |  
    a custom: b ].  
B transmit to: #details;  
from: #nav port: #class;  
andShow: [:a |  
    a text  
        display: #sourceText ].  
B transmit to: #details;  
from: #nav port: #method;  
andShow: [:a |  
    a text  
        display: #sourceText ].
```

One picture is worth one thousand words

Which one?

How could it be that simple?



Program visualization is difficult

Limited number of colors: 12

Blur and color emergence

Limited screen size

Limited context, edges crossing

Limited short-term memory (three to nine)

Difficult to remember too many symbols/semantics

Culture, Colorblind

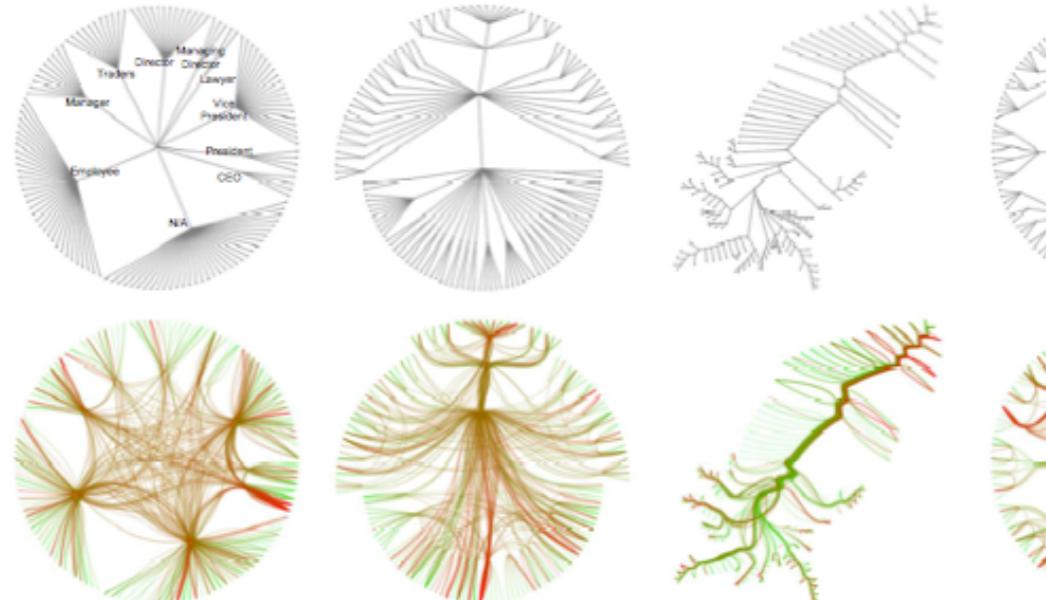
Our constraints

Lot of existing and advanced solution:

ICPC is full of them

HEB

Plenty of works on information visualization



Simple but not simplistic

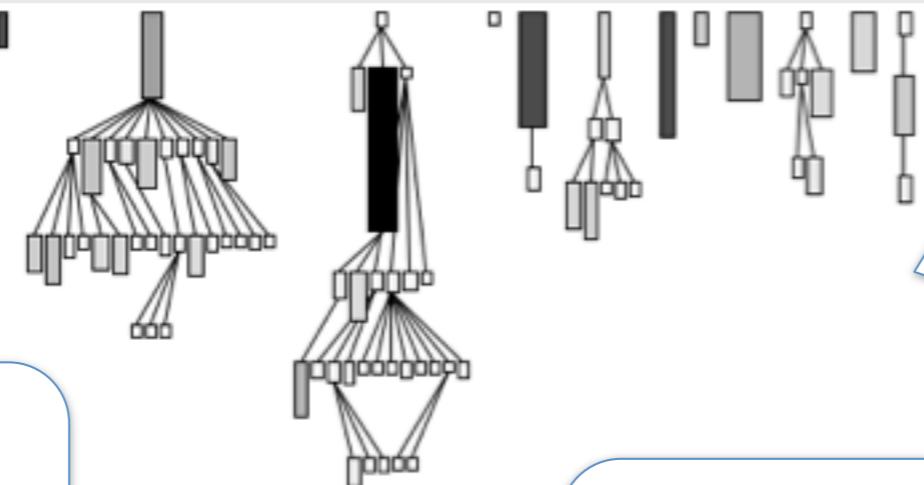
Ideally, solutions that an engineer could reproduce in a couple of days

Understanding large systems

- Understanding code is difficult!
 - Systems are large
 - Code is abstract
 - Should I really convinced you?
-
- Some existing approaches
 - Metrics: you often get meaningless results once combined
 - Visualization: often beautiful but with little meaning
 - Polymetric view is an idea of M. Lanza [WCSE,TSE]

Class Blueprint

Outer Rectangles = Classes
Inner = methods/attributes
Demonstrates class layers and class interactions



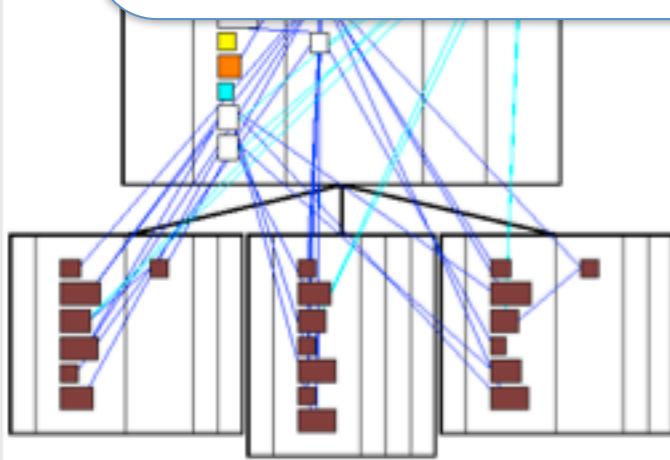
System Complexity

Classes and their hierarchy
Height = NOM
Width = NOA
Color = LOC



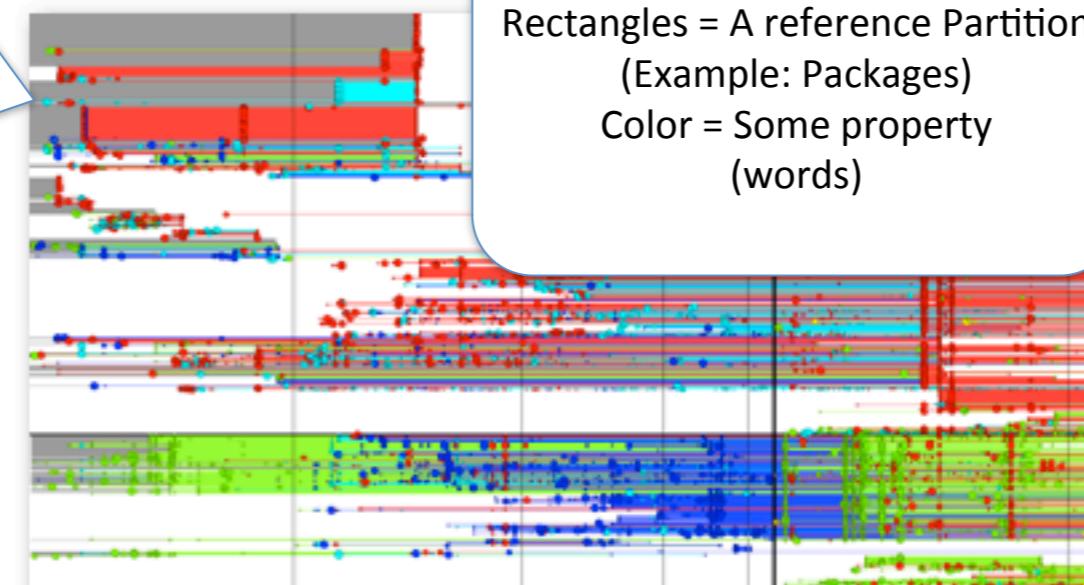
Chronia

X-axis = time
Y-axis = files
Colors = authors



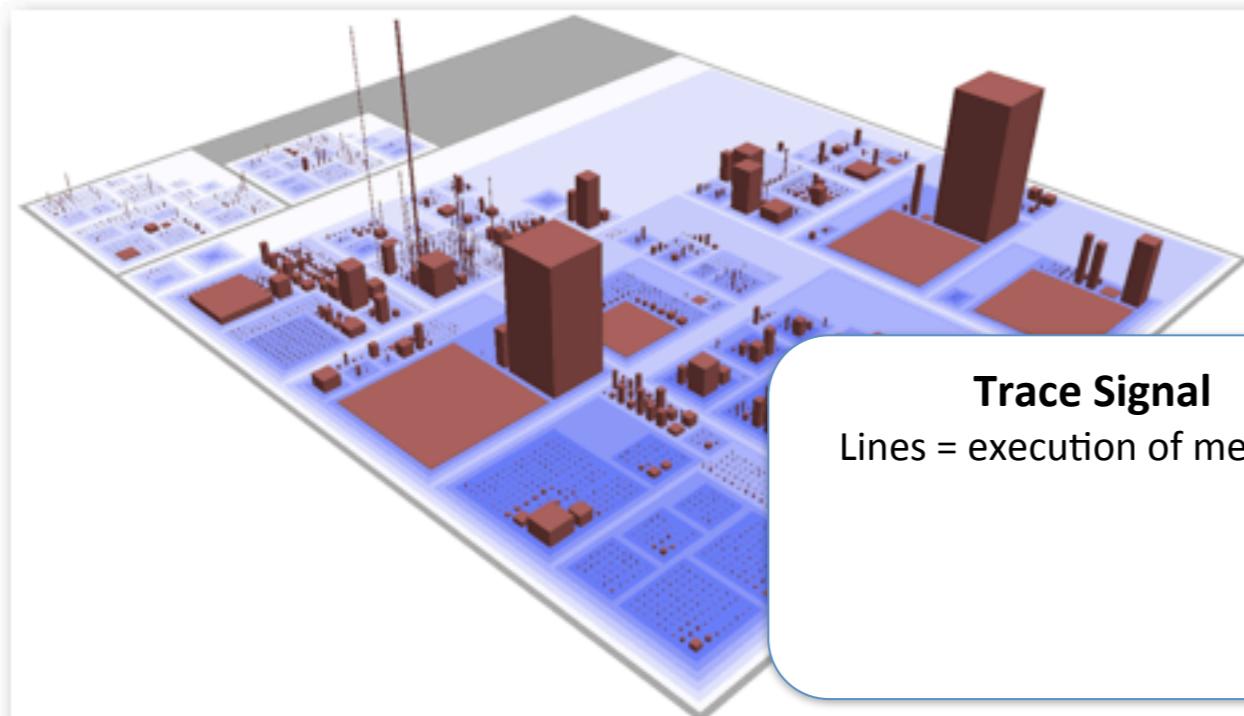
Distribution Map

Rectangles = A reference Partition (Example: Packages)
Color = Some property (words)



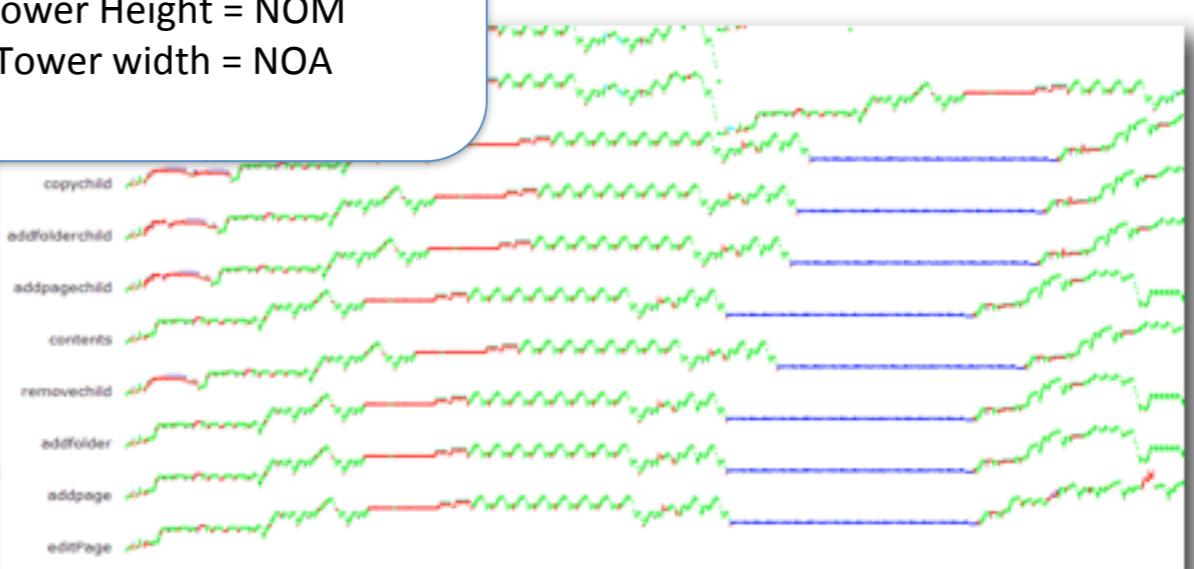
Code City

Districts = Packages
Towers = Classes
Tower Height = NOM
Tower width = NOA

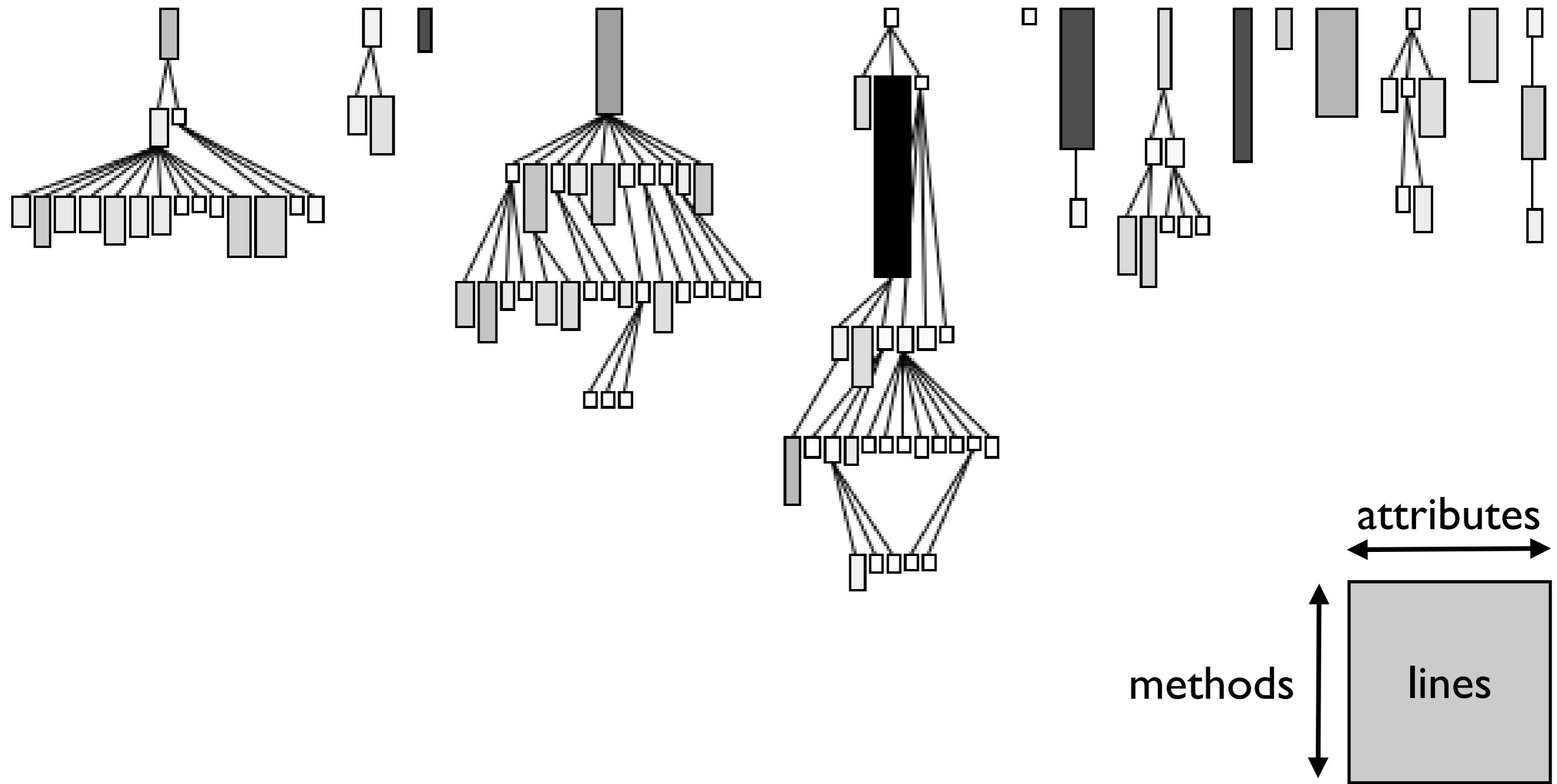


Trace Signal

Lines = execution of methods

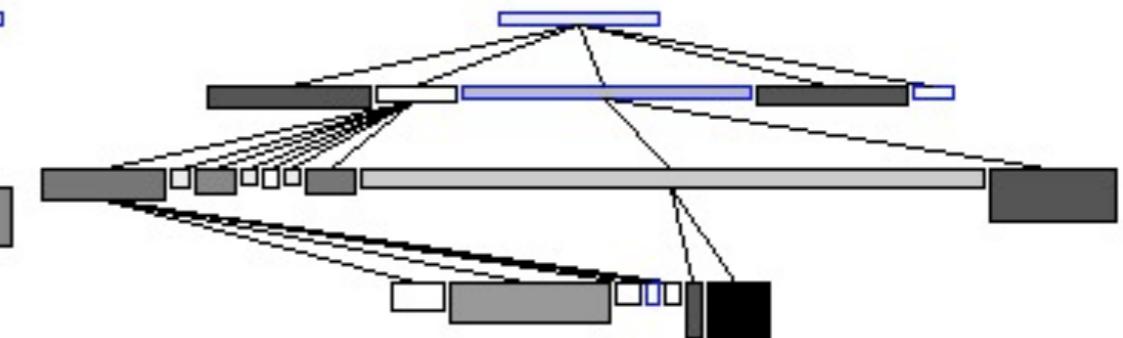
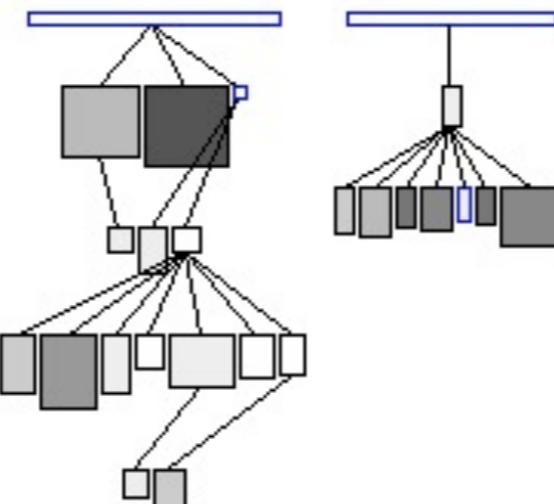
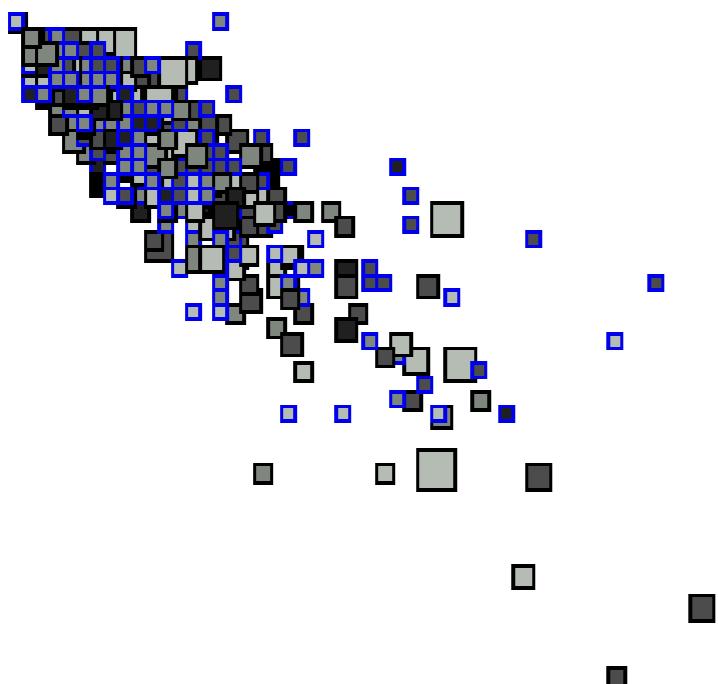


System Complexity shows class hierarchies.



Polymetric views condense information

To get a feel of the inheritance semantics: adding vs. reusing



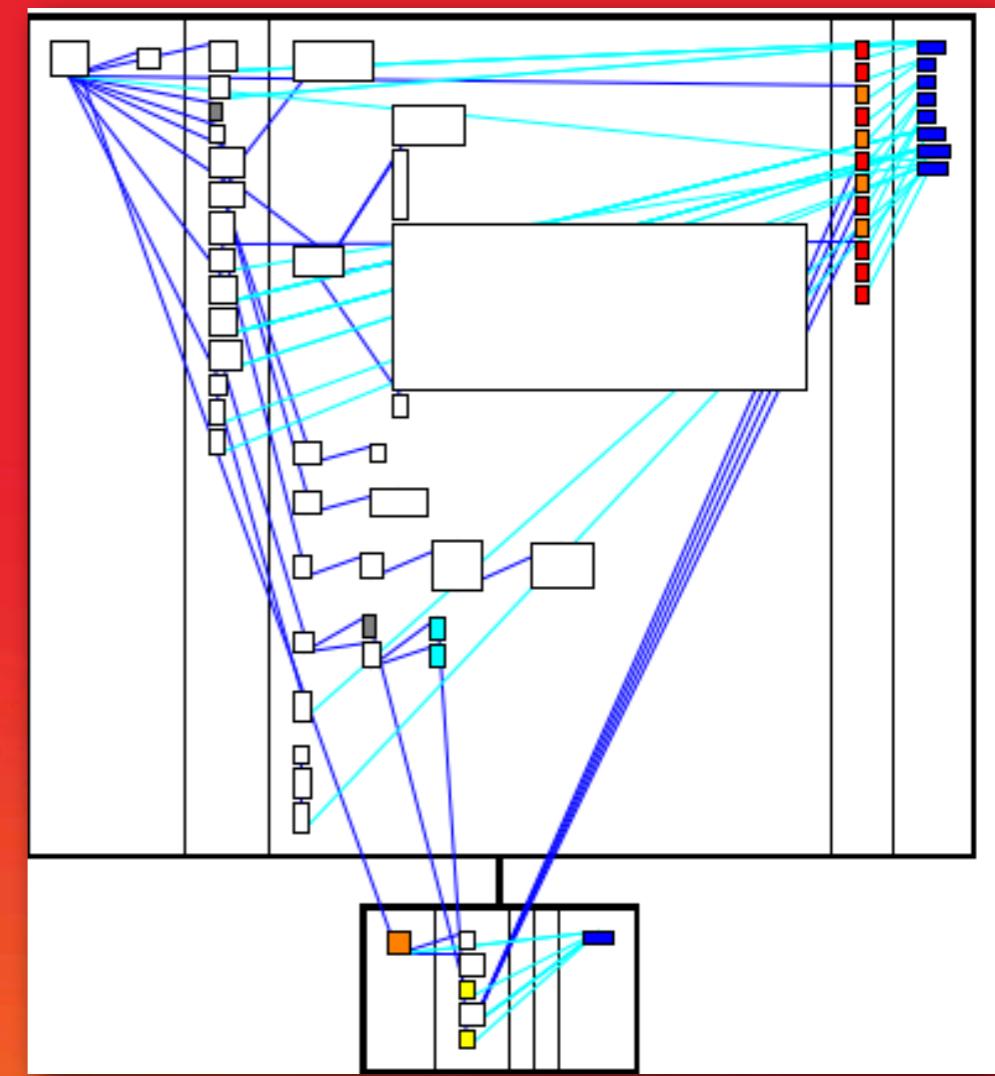
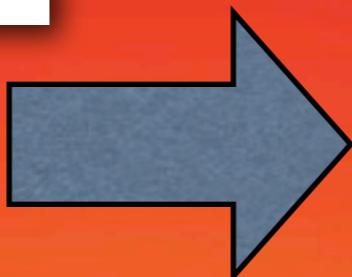
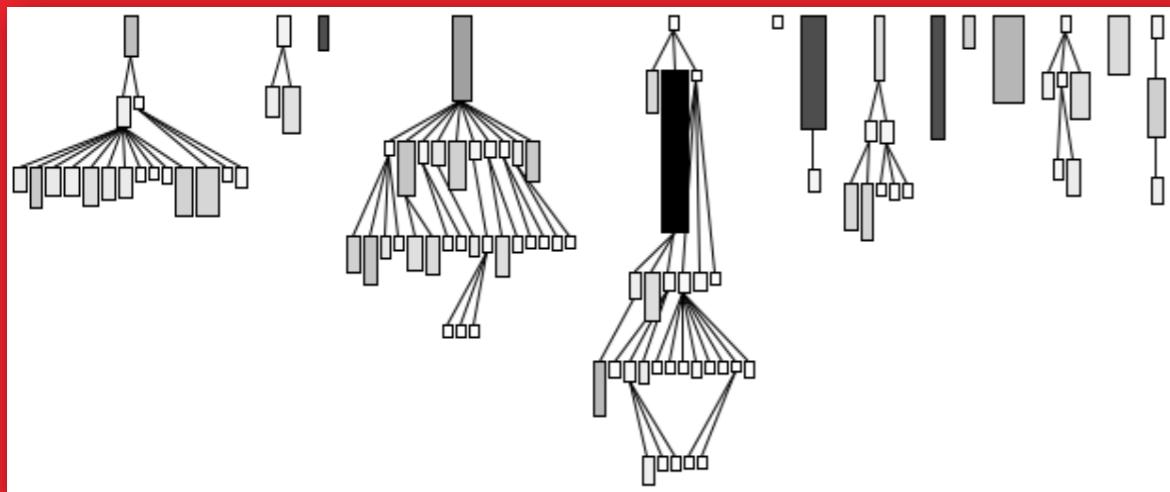
Classes+Inheritance

- W: # of Added Methods
- H: # of Overridden Methods
- C: # of Method Extended

methods
LOC
statements
parameters

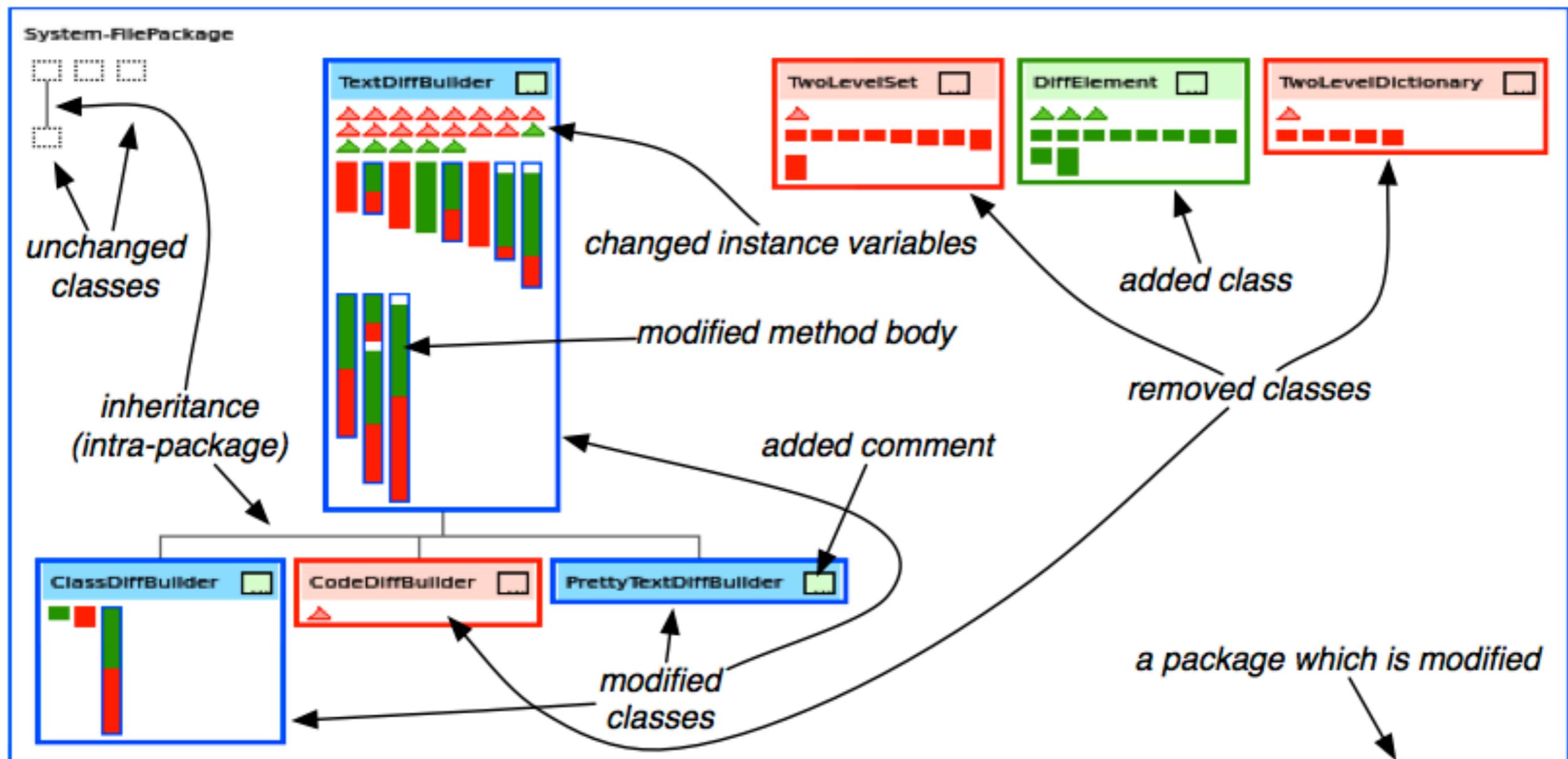
Understanding Classes: Easier?

- Public and non public methods
- No predefined reading order
- Inheritance

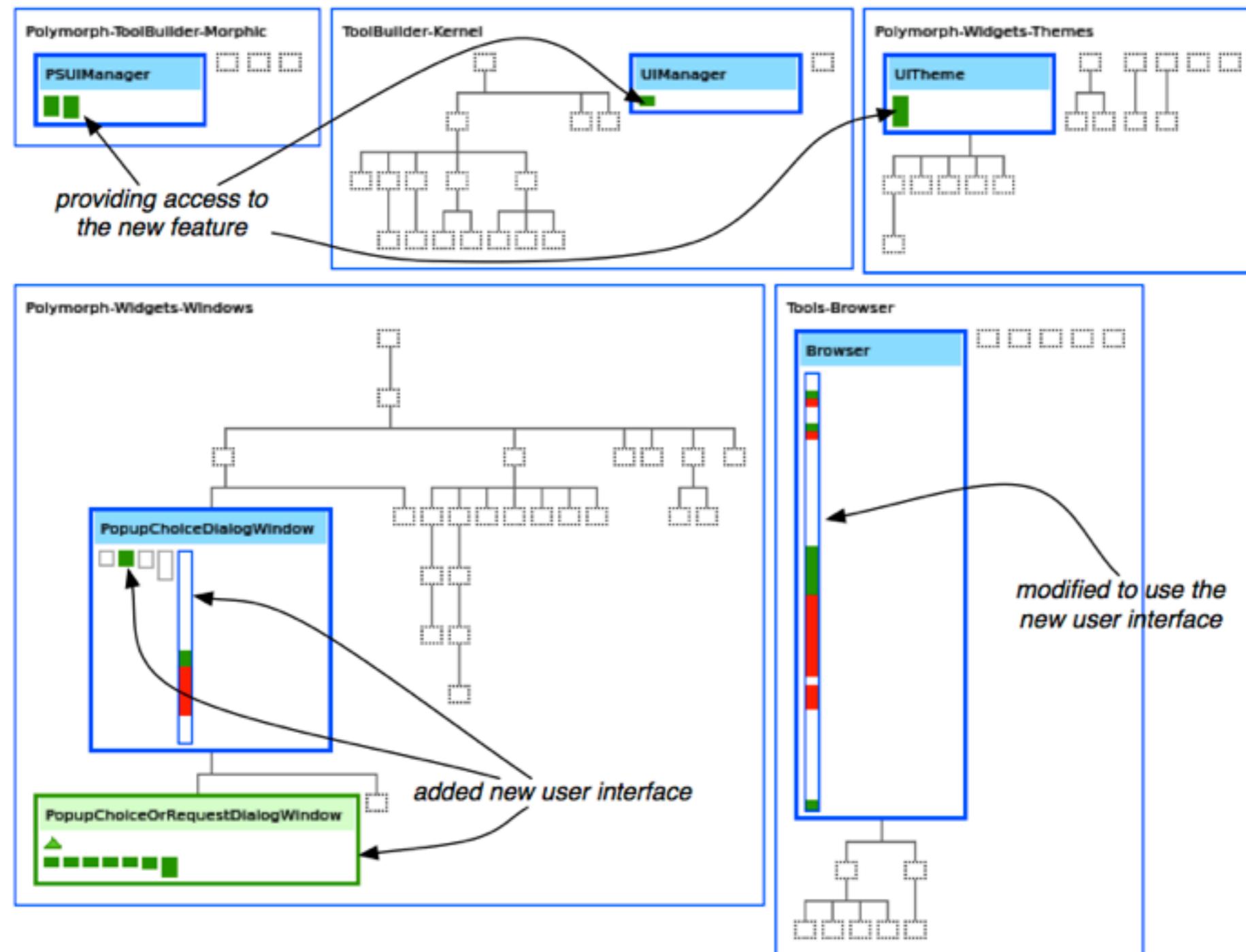


- Class blueprint is an idea of M. Lanza [OOPSLA]

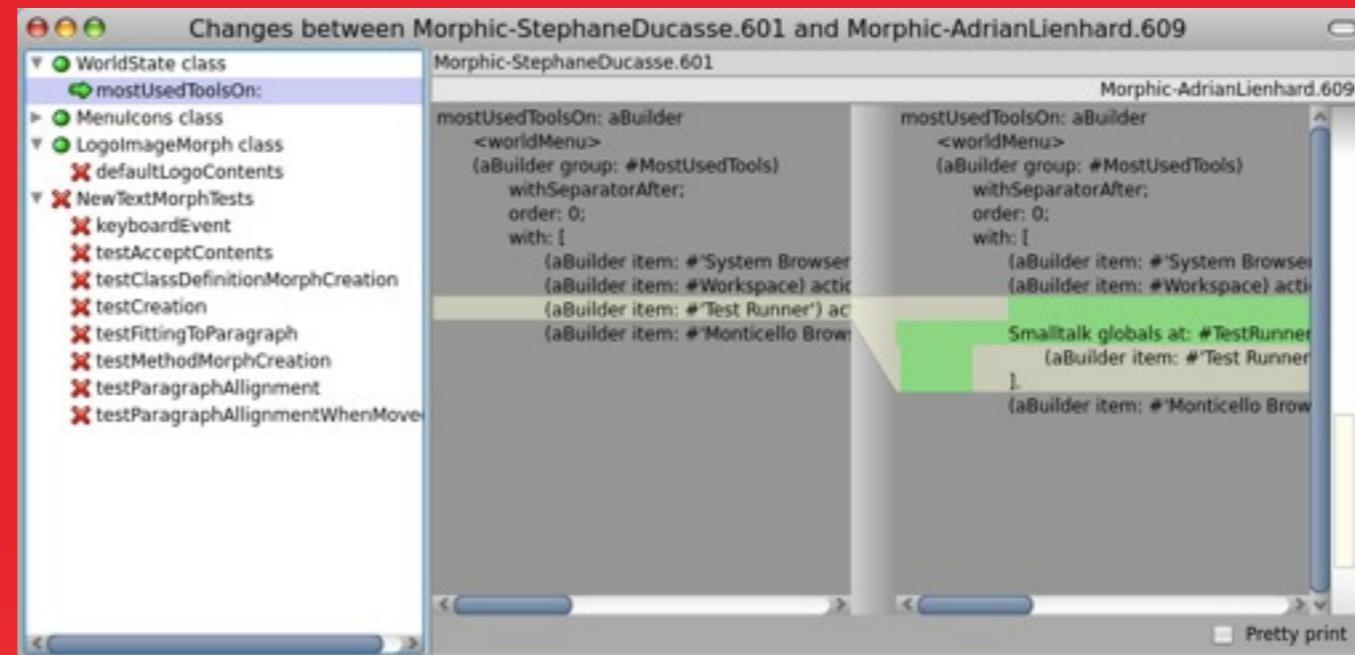
Understanding Changes



Introducing a feature



How to understand changes



```
Changes between Morphic-StephaneDucasse.601 and Morphic-AdrianLienhard.609

Menulcons class >>radioButtonPressedIconContents (removed)
Menulcons class >>rotateIconContents (removed)
Menulcons class >>scaleIconContents (removed)
Menulcons class >>solidMenuIconContents (removed)
Menulcons class >>tinyMenuIconContents (removed)
NewTextMorphTests (removed)
NewTextMorphTests >>keyboardEvent (removed)
NewTextMorphTests >>testAcceptContents (removed)
NewTextMorphTests >>testClassDefinitionMorphCreation (removed)
NewTextMorphTests >>testCreation (removed)
NewTextMorphTests >>testFittingToParagraph (removed)
NewTextMorphTests >>testMethodMorphCreation (removed)
NewTextMorphTests >>testParagraphAlignment (removed)
NewTextMorphTests >>testParagraphAlignmentWhenMoved (removed)
WorldState class >>mostUsedToolsOn: (changed)

mostUsedToolsOn: aBuilder
<worldMenu>
(aBuilder group: #MostUsedTools)
withSeparatorAfter;
order: 0;
with:
(aBuilder item: #'System Browser')
(aBuilder item: #'Workspace') action:
(aBuilder item: #'Test Runner') action:
(aBuilder item: #'Monticello Brow
Smalltalk globals at: #TestRunner
(aBuilder item: #'Test Runner')
].
(aBuilder item: #'Monticello Brow

Pretty print
```

- Torch is the work of V. Uquillas-Gomez

Torch Dashboard: Changes from SLICE-Issue1709-EnhancedTextDiffBuilder (ancestor) to SLICE-Issue1709-EnhancedTextDiffBuild

Change Summary

Packages (32) 3
 Classes (149) 2 4 3
 Methods (1646) 25 8 39
 Variables (157) 9 18
 additions: 9
 removals: 18

Colors Borders

Added	Removed	Modified	
			Comment
			N/A
			Method source
N/A	N/A		Class' package

Viz. Class Status Viz. Width
 All changed 900

Viz. Relationships
 intra- packages

Classes / Methods

- PrettyTextDiffBuilder
- TextDiffBuilder
 - attributesOf:
 - buildDisplayPatch
 - buildDisplayPatchFrom:to:
 - buildDisplayPatchFrom:to:inC
 - buildDisplayPatchFrom:to:inC
 - buildPatchSequence
 - buildReferenceMap
 - collectRunFrom:startingWith:
 - destString:
 - detectShiftedRuns
 - findMatches
 - formatLine:
 - from:to:

Changed Packages (details) Changed Packages Packages Changed Classes (details) Classes Symbolic Clouds

Tests-System

Settings-Tools

System-FilePackage
 class method Protocol: instance creation Author: HenrikSperreJohansen
 buildDisplayPatchFrom: sourceText to: destinationText
 ^ (self from: sourceText to: destinationText) buildDisplayPatch
 buildDisplayPatchFrom: srcString to: dstString
 ^ (self from: srcString to: dstString) buildDisplayPatch

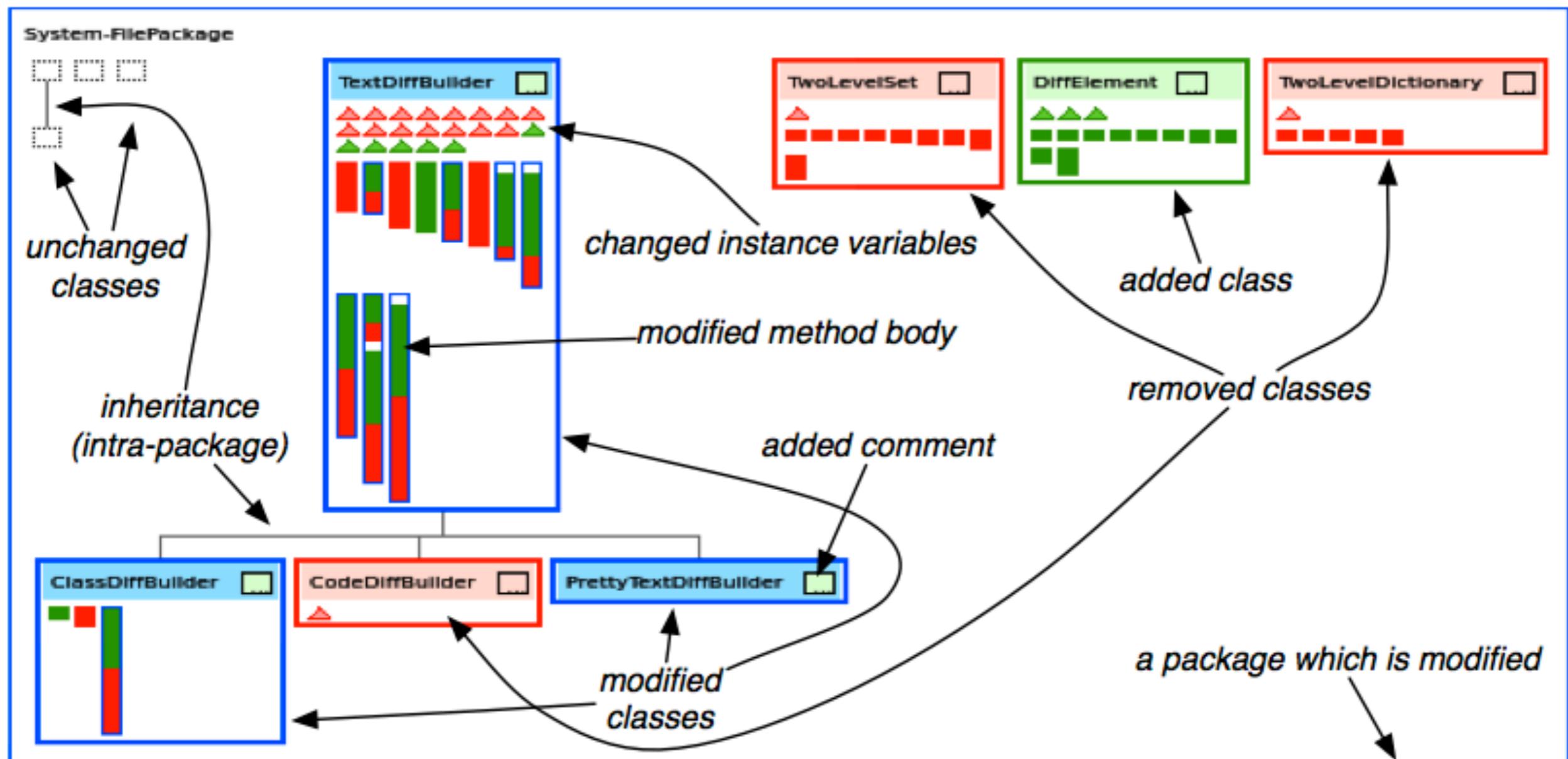
TextDiffBuilder
 ClassDiffBuilder
 CodeDiffBuilder
 PrettyTextDiffBuilder

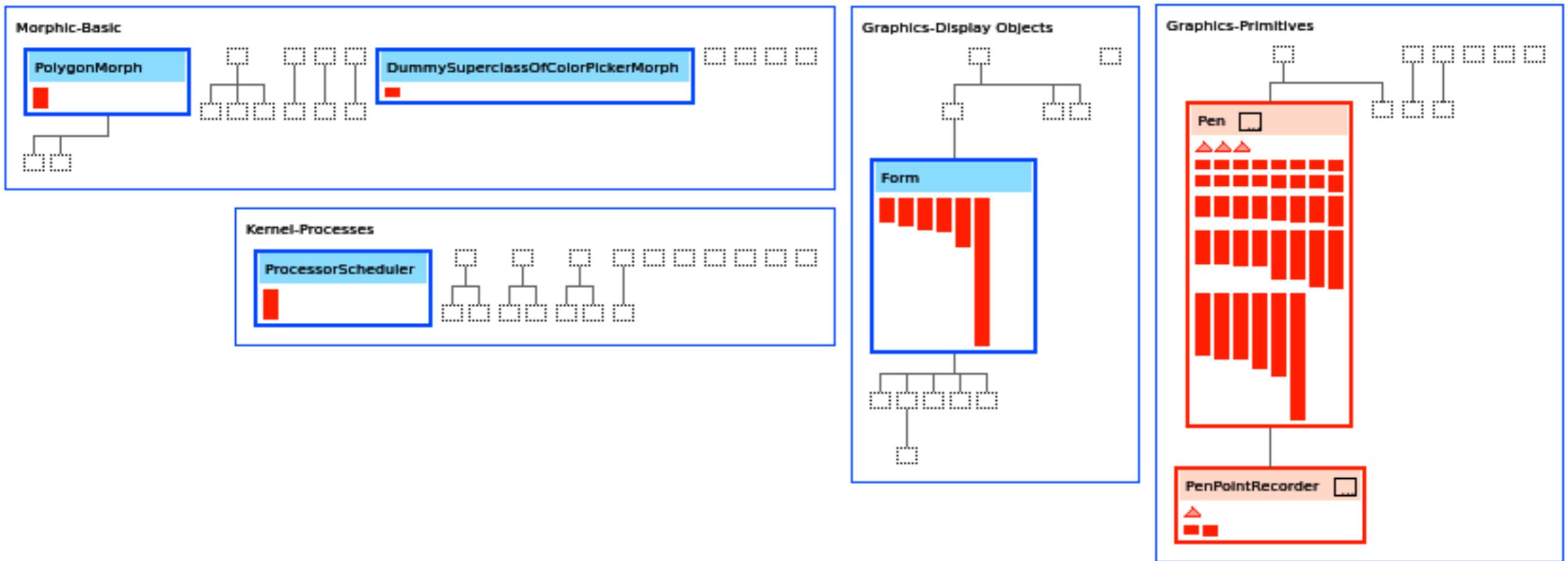
Source Code Diff Source Code Protocol/Author

buildDisplayPatch
 ^ Text streamContents:[:stream]
 self printPatchSequence: self buildPatchSequence on: stream
]

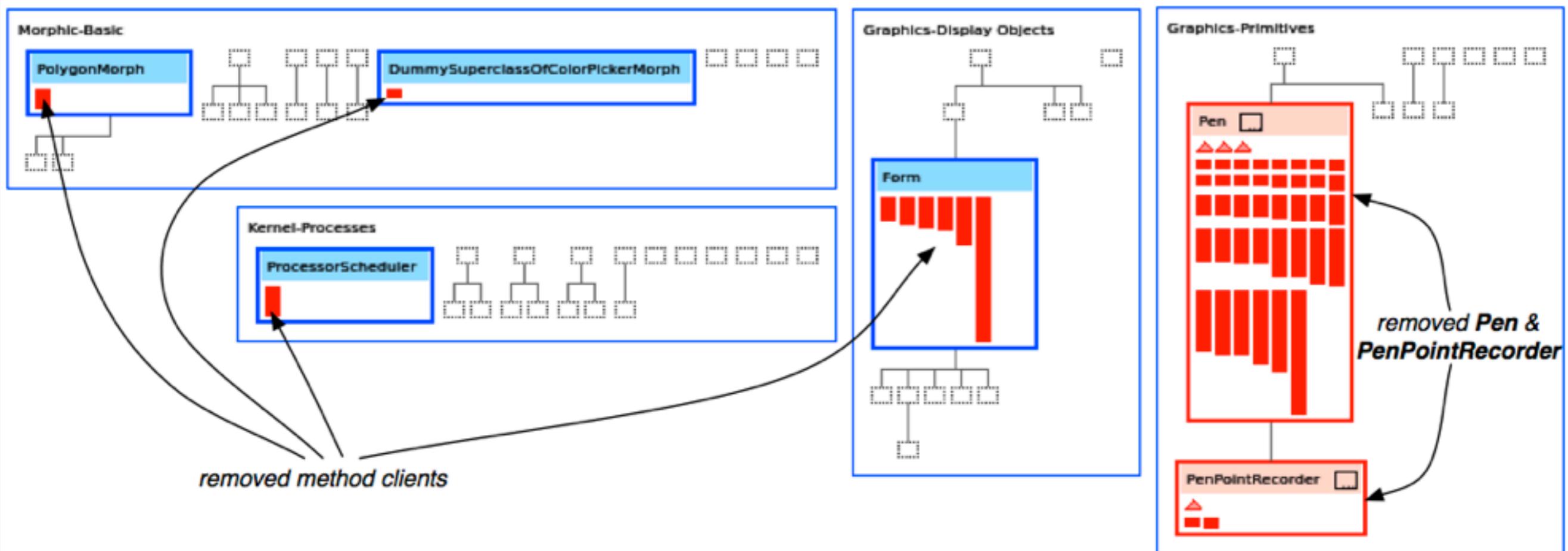
buildDisplayPatch
 ^ Text streamContents: [:stream]
 self
 patchSequenceDolfMatch: [:string |
 self print: string withAttributes: NormalTextAttributes
]
 ifInsert: [:string |
 self print: string withAttributes: InsertTextAttributes
]
 ifRemove: [:string |
 self print: string withAttributes: RemoveTextAttributes
]
]

Package Structure





Removing a feature (I)



Editing comments



Lessons learned

- Program visualization is difficult
- Squares and little symbols are just squares and little symbols
- Glancing at code is still efficient

Omnipresent code + visualization is excellent

The screenshot shows a Smalltalk environment with the following details:

- SystemNavigation**: The modified class.
- method showing addition and removal**: A specific method within the class.
- instance method Protocol: browse Author: AndyKellens**: Metadata for the method.
- Method code:**

```
browseAllObjectReferencesTo: anObject except: objectsToExclude ifNone: aBlock
"Bring up a list inspector on the objects that point to anObject.
If there are none, then evaluate aBlock on anObject.

| aList shortName |
aList := anObject pointersToExcept: objectsToExclude.
aList := PointerFinder pointersTo: anObject except: objectsToExclude.
aList size > 0 ifFalse: [^aBlock value: anObject].
shortName := (anObject name ifNil: [anObject printString]) contractTo: 20.
aList inspectWithLabel: 'Objects pointing to ', shortName
```
- Annotations:**
 - A callout points to the class name with the text *modified class*.
 - A callout points to the method name with the text *method showing addition and removal*.
 - A callout points to the metadata bar with the text *scope, protocol and author of method*.
 - A callout points to the method code with the text *method diff as a fly-by-help*.



Dedicated tools tailored to your problems

Profitable in terms of cost

<http://www.synectique.eu>

Inventive Toolkit

- Multi-language



- Meta-tool: An environment to build tools



- Any kind of data: logs, authors, bugs,

...

Example of dedicated tools

synectique
Inventive Analysis

<http://www.synectique.eu>

Analysis and Migration Support

Problem: Since 30 years company X develops insurance solution. The old compiler costs more and more.

Which part to migrate first?

How to reduce the migration cost (duplicated code, screen numbers)?

How to control the migration?

Solution :

Build a specific analysis tools **(2 cycles of 6 weeks)**

Domain and problem analysis

Engineer formation

Reverse engineering and support new generation

Problem: Since 15 years company X develops missile shooting systems. The most of original team left. Other foreign code to maintain.

Understanding current message driven architecture.

Extracting domain model

Mapping concepts back to code.

Supporting testing components

Solution :

Domain and problem analysis

Build a specific analysis tools (3-4 months)

Engineer formation

Decision Making Support

Problem: Since 20 years company X develops insurance solution. The old compiler costs more and more.

Which part depends on which other one?

Duplicated code?

Dead code?

Solution :

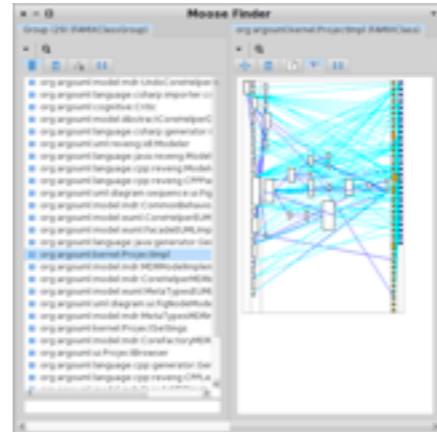
Domain and problem analysis

Build a specific analysis tools (2 cycles of 6 weeks)

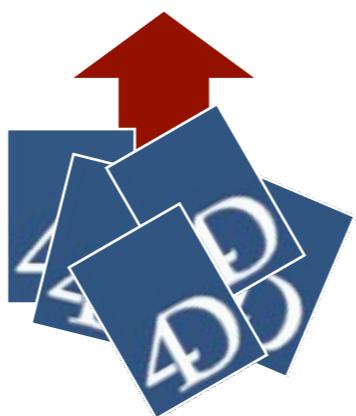
Engineer formation

classes select: #isGod

McCabe = 21c = 753,000

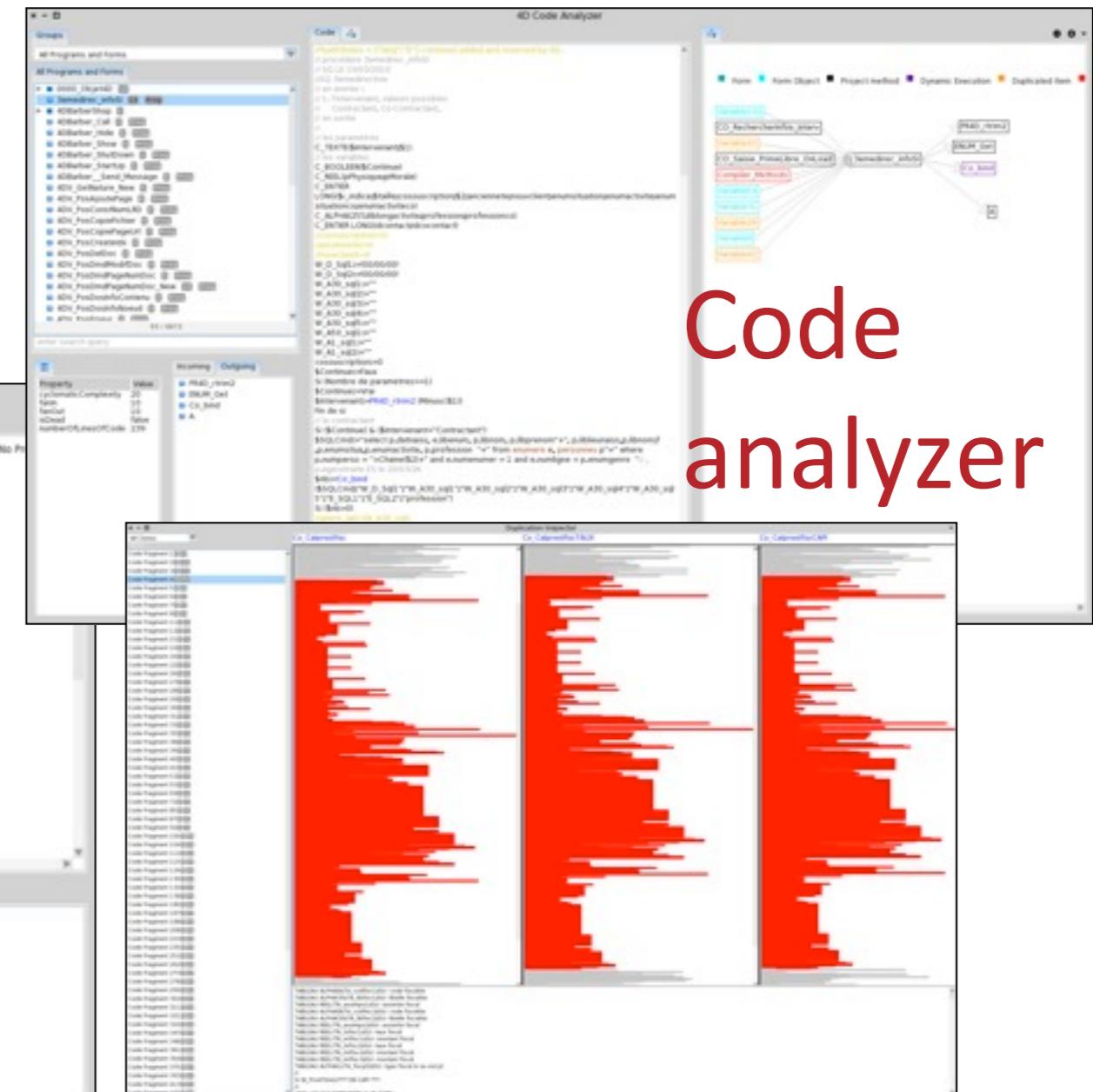
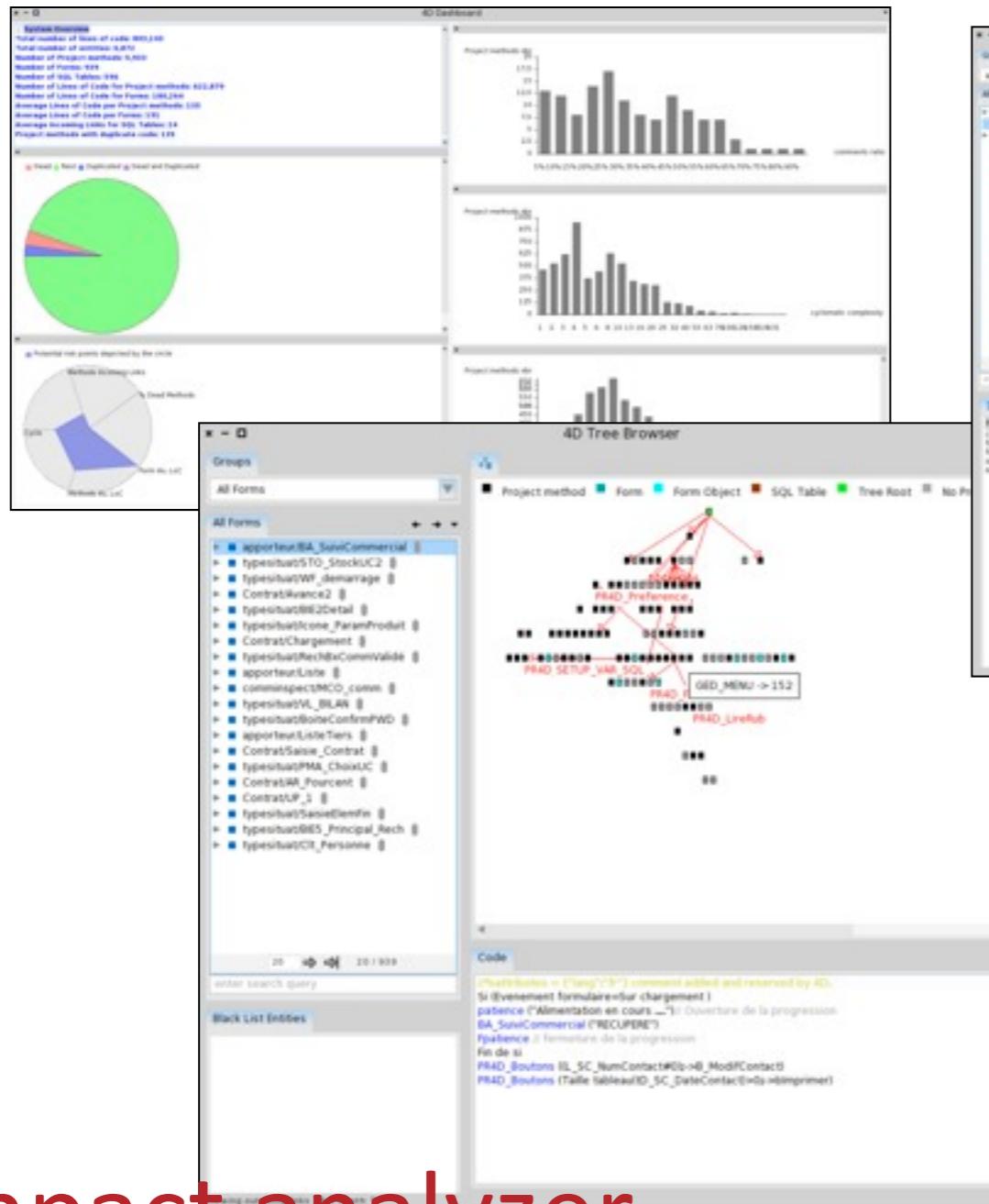


Inventive Toolkit



Multi levels

Dashboard



Impact analyzer

Code
analyzer

Duplication
inspector

Dashboard



A screenshot of the 4D Dashboard application window. On the left, there's a 'System Overview' panel with a list of statistics: Total number of lines of code: 803,143, Total number of entities: 6,872, Number of Project methods: 5,933, Number of Forms: 939, Number of SQL Tables: 596, Number of Lines of Code for Project methods: 622,879, Number of Lines of Code for Forms: 180,264, Average Lines of Code per Project methods: 135, Average Lines of Code per Forms: 191, Average Incoming Links for SQL Tables: 24, and Project methods with duplicate code: 135. Below this is a pie chart divided into four segments: Dead (red), Rest (green), Duplicated (blue), and Dead and Duplicated (purple).

On the right side of the dashboard, there are three stacked bar charts. The top chart shows 'Project methods nbr' on the y-axis (0 to 20+) against a 'comments ratio' x-axis (5% to 70%). The middle chart shows 'Project methods nbr' on the y-axis (0 to 1000) against a x-axis with categories 1, 2, 3, 4, 5, 6, 8, 10, 13, 16, 20, 25, 32, 40, 50, 63, 78, 0, 112, 58. The bottom chart shows 'Project methods nbr' on the y-axis (0 to 650) against a x-axis with categories 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100.

To the right of the dashboard, a list of features is presented:

- Metrics:
 - Method projects
 - Forms
 - Tables
- Quality
- Debt
- Reports

Code Analyzer



4D Code Analyzer

Groups

All Programs and Forms

All Programs and Forms

- 0000_Objet4D
- 3medirec_info\$1
- 4DBarberShop
- 4DBarber_Call
- 4DBarber_Hide
- 4DBarber_Show
- 4DBarber_ShutDown
- 4DBarber_Startup
- 4DBarber_Send_Message
- 4DV_GoToNature_New
- 4DV_PosAjoutePage
- 4DV_PosConstructurRD
- 4DV_PosCopierFichier
- 4DV_PosCopiePageUrl
- 4DV_PosCreatedIdx
- 4DV_PosDelDoc
- 4DV_PosDmdModifDoc
- 4DV_PosDmdPageNumDoc
- 4DV_PosDosInfoContenu
- 4DV_PosDosInfoNoeud
- 4DV_ProfFraise

enter search query:

Incoming Outgoing

Property	Value
cyclomaticComplexity	20
fanIn	10
fanOut	10
isDead	false
numberOfLinesOfCode	239

Code

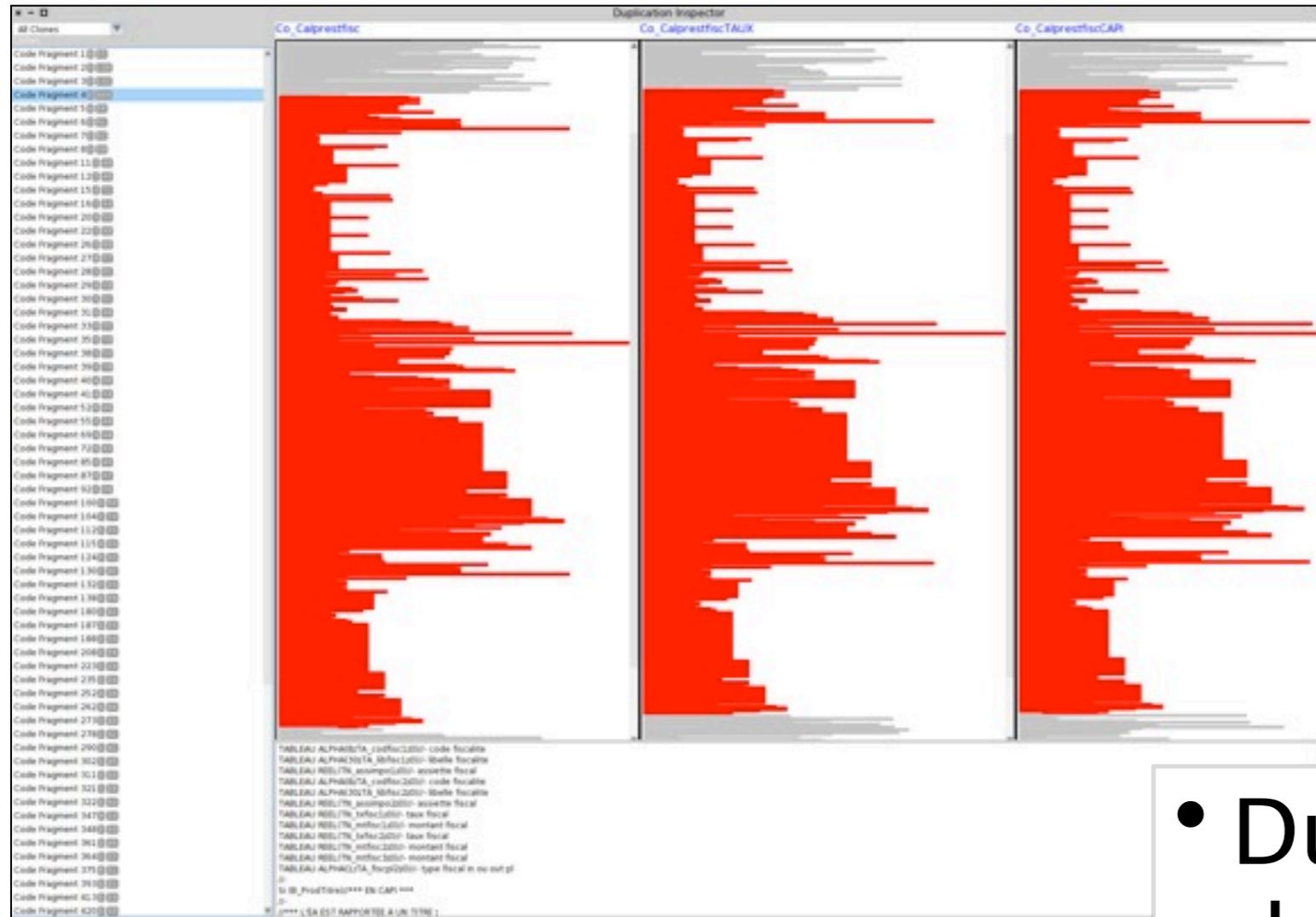
```
/*#Attributes = $Lang("PR") comment added and reserved by 4D;
$ procédure 3medirec_info$1
$SQL LE 19/03/2010
$SQL 3medirec
$ en entrée:
$ 1. l'intervenant, valeurs possibles
$   - Contractant, Co-Contractant,
$ en sortie:
$ les paramètres
C_TEXTE$1(Intervenant$1)
C_BOOLEEN$1(Contractuel)
C_RELIP(PhysiqueMorale)
C_ENTIER
LONG($1_indexe$taillecosouscription$2ancienneteanouveauclientenumsituationenumactiviteenum
situationconsumactifcole
C_ALPHA(255)longactiviteprofessionprofessionco)
C_ENTIER LONG(contactidcontact)
$0$description$0
$anciennete$0
$consumactifcole$0
$A1(sql1:=
W_D_Sql1=00/00/00
W_D_Sql2=00/00/00
W_A30_sql1:=
W_A30_sql2:=
W_A30_sql3:=
W_A30_sql4:=
W_A30_sql5:=
W_A50_sql1:=
W_A1_sql1:=
W_A1_sql2:=
cosouscription<0
$Contractuel=Faux
Si (Nombre de paramètres>=1)
$Continue=Vrai
$Intervenant=$PR4D_rtrim2 (Minusc($1))
Fin de si
Si le contractant
Si ($Contractuel & ($Intervenant="Contractant"))
$SQLCmd="select p.datnaiss, e.libelnum, p.libnom, p.libprenom+", p.libleunaiss,p.libnom/
,p.unumsitua,p.unumactifcole, p.profession "+$ from enumere e, personnes p"+ where
p.unumperso = "+Chaine($2)+" and e.unumerenumer = 1 and e.unumligue = p.unumligue "+"
$p.unumligue ES le 29/03/2K
$nb=$Co_bind
($SQLCmd+W_D_Sql1" W_A30_sq1" "W_A30_sq2" "W_A30_sq3" "W_A30_sq4" "W_A30_sq5"
" "TTE_SQL1" "TTE_SQL2" "profession")
Si ($nb>0)
    $prenom=$Co_bind(W_A30_sq1)
    NumPerso_Jabz=$2
    nom_Jabz=W_A30_sq2
    prenom_Jabz=W_A30_sq3
    nomf_Jabz=W_A30_sq5
    Dnaiss_Jabz=W_D_Sq3
    Lieunaiss_Jabz=W_A30_sq4
    pPhysique=1
    pMorale=0
$SQLCmd=" select case when pdatcrea is null then '2009-1-01' else datcrea end from
```

Form Form Object Project method Dynamic Execution Duplicated Item

The dependency graph on the right side of the interface shows various nodes representing variables and objects, connected by lines indicating dependencies. Nodes include 'Variable192', 'CD_RechercherInfo_interv', 'Variable42', 'CD_Saisie_PrimeLibre_OnLoad', 'Compiler Methods', 'PR4D_rtrim2', 'ENUM_Get', 'Co_bind', and several unnamed variable nodes labeled 'Variable 4', 'Variable 11', 'Variable 29', 'Variable 9', and 'Variable 42'. The connections show how these elements interact within the code logic.

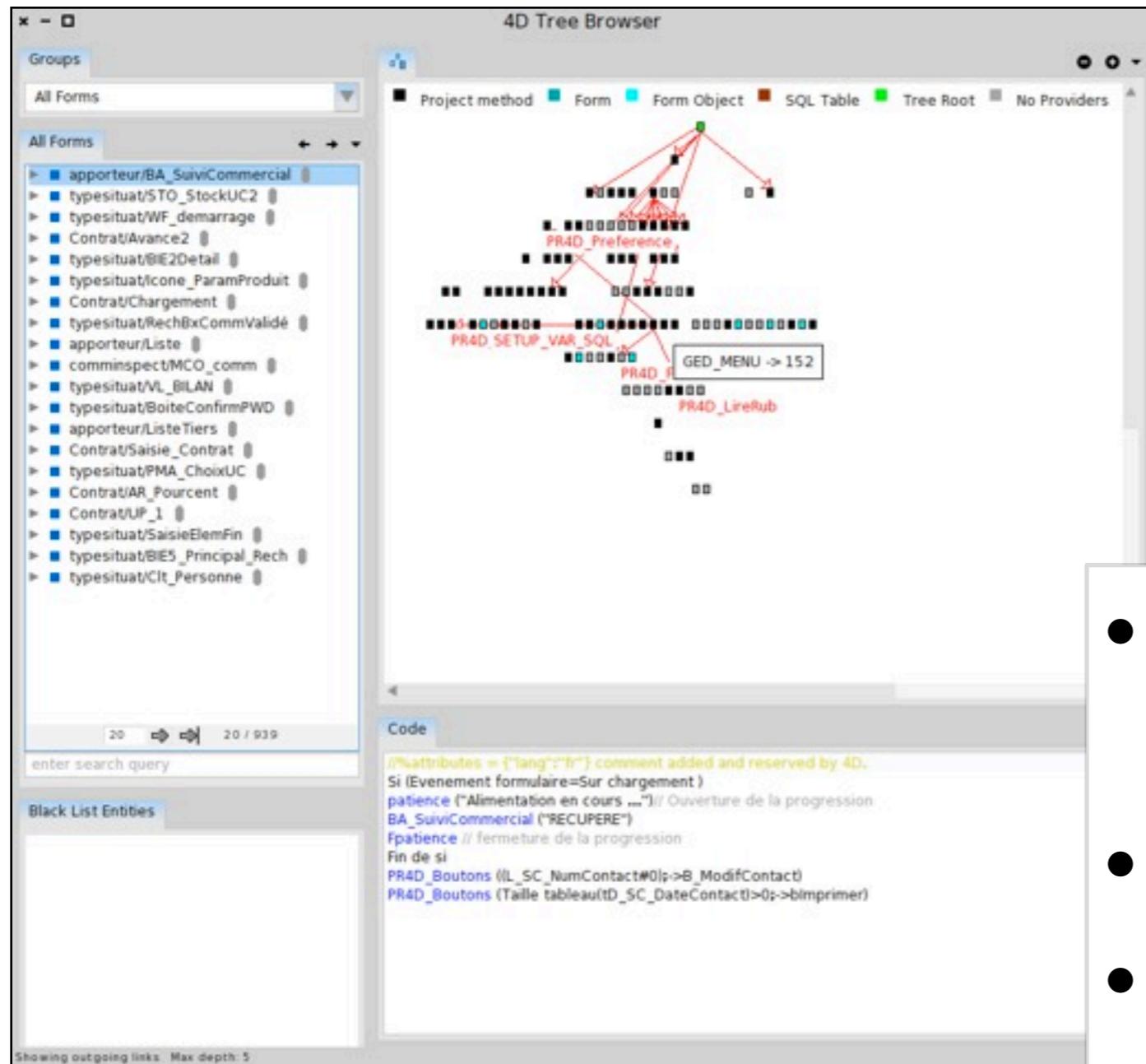
- Code Browser
- Dependencies
- Filters

Duplication Inspector



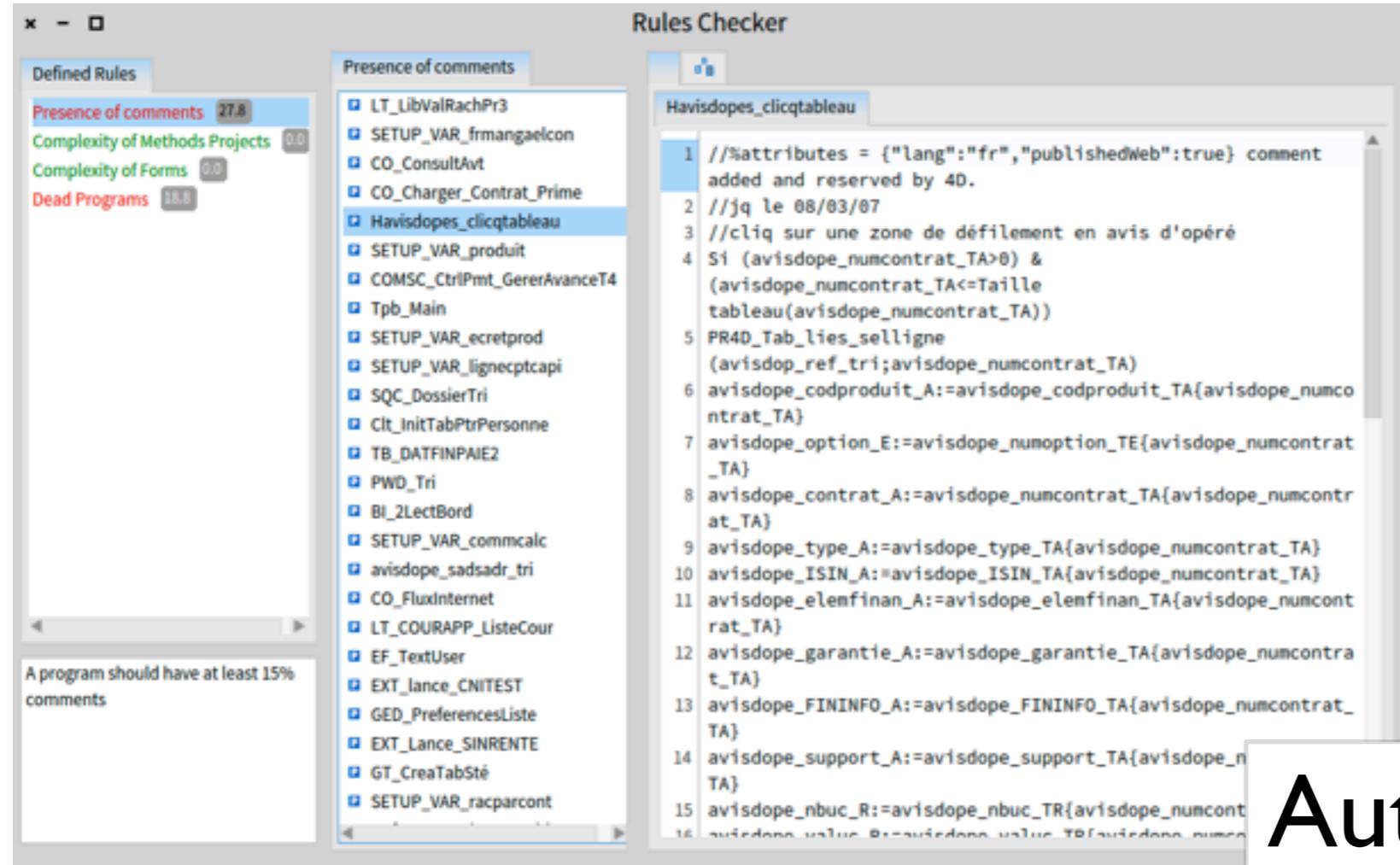
- Duplication detector
- Customizable

Impact Analyzer

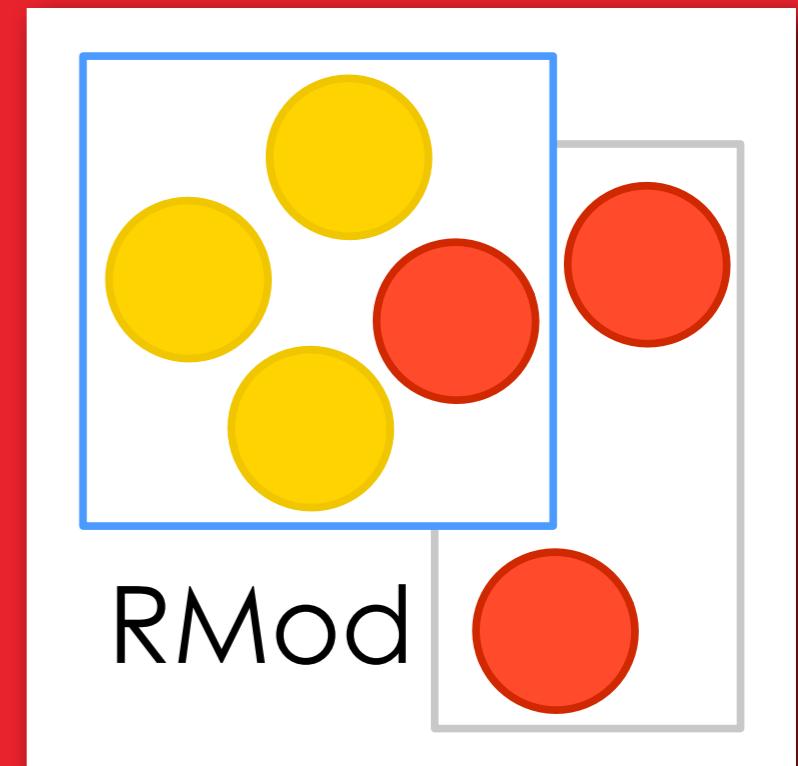


- Global dependencies
- Interactif
- Cyclic dependencies

Rule Checker



Automated rule
checking
Specific client driven
rules



<http://stephane.ducasse.free.fr>

<http://www.synectique.eu>

Interested by your problems

- Migration support
- Decision making support
- Extraction and definition of rules
- Software architecture Verification
- Visualizations
- Cost prediction
- Impact change
- Service-oriented architecture
- Software analysis

