

Code quality assessment tools

Alexandre Bergel & Romain Robbes

abergel@dcc.uchile.cl

06/10/2010

Outline

Enriched DSM

Package Blueprint

Distribution map

Lint

Kai Profiler

Hapao Test coverage

Enriched DSM

Problem

Software composed of many different packages are likely to contains dependency cycles. Those cycles are problematic since they increase the complexity of the application

More info

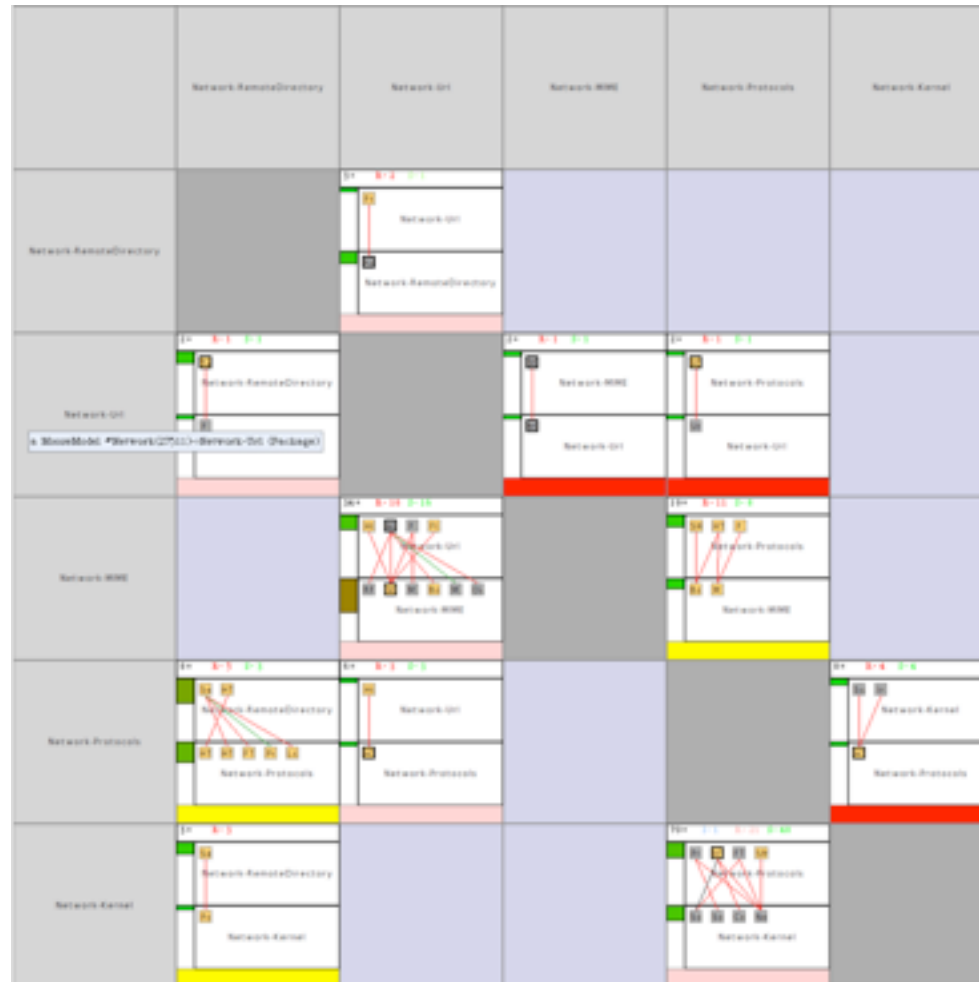
<http://www.moosetechnology.org/tools/edsd>

<http://www.jannik-laval.eu/assets/files/papers/Lava09c-WCRE2009-eDSM.pdf>

Enriched DSM

	Net	Net	Net	Net	Net	Net	Net	Net	Net	Net
Net										
Net										
Net			154							
Net			30							
Net										3
Net					3			70		
Net	4			1				19	34	
Net	2	15			8	8			6	
Net	4	51	1		2		2	2		
Net		2							2	6

Enriched DSM

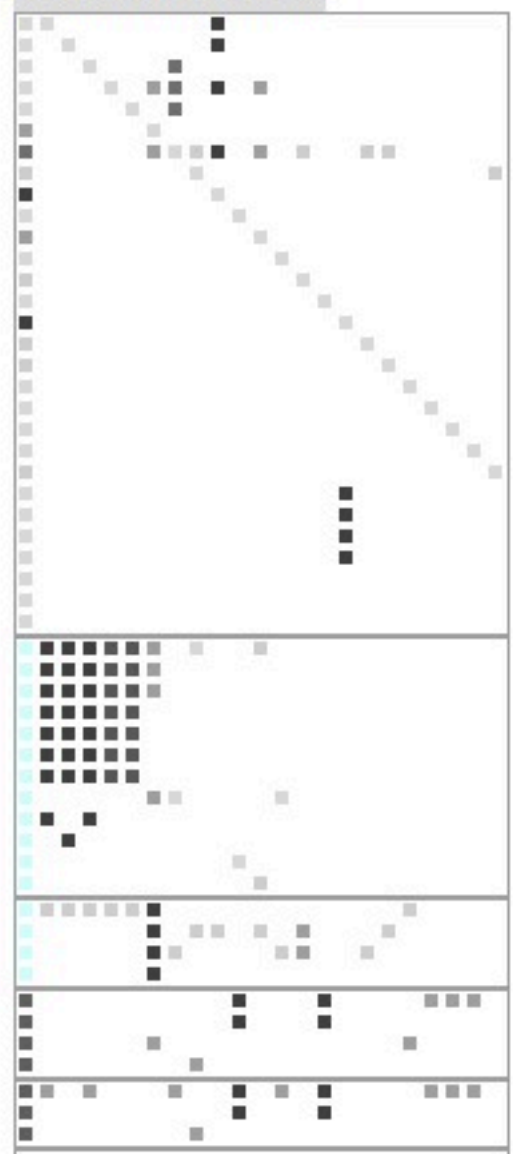


Package Blueprint

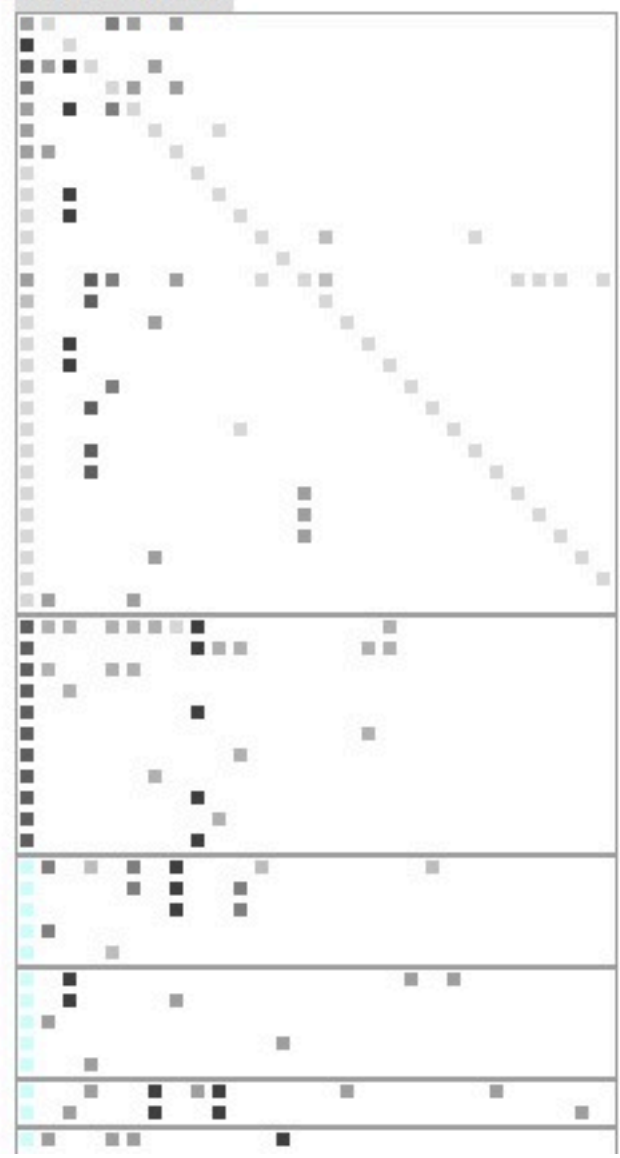
Problem

Understanding how packages are structured in general and how they relate each other

Metacello-TestsMC



Metacello-MC



Metacello-Core



Metacello-TestsCore



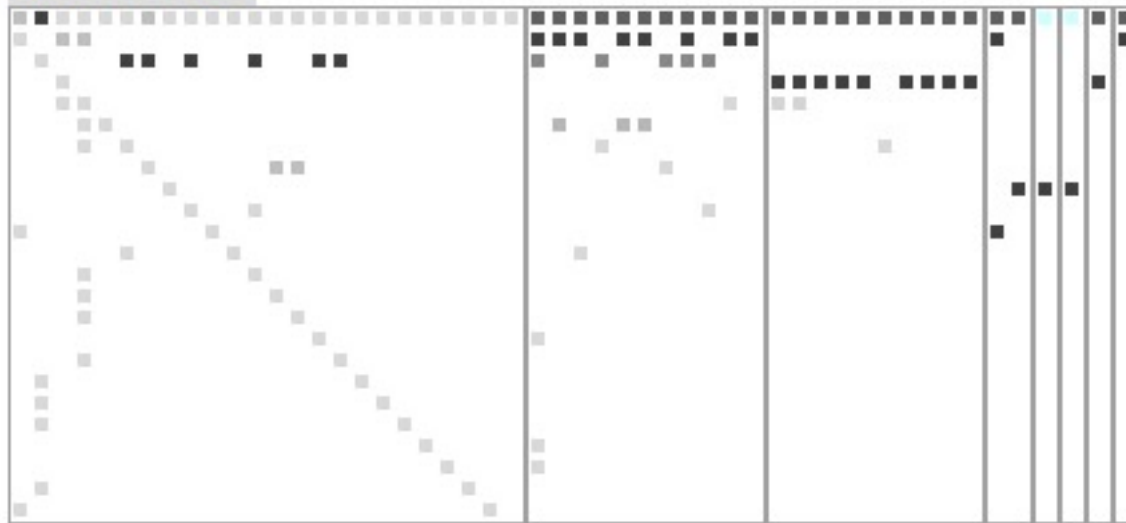
Metacello-Reference



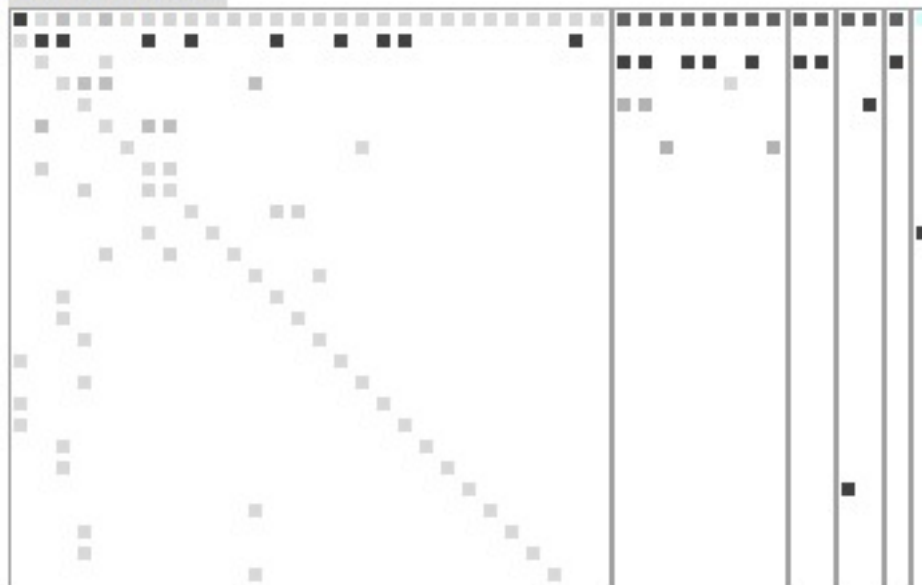
Metacello-Platform



Metacello-Core



Metacello-MC



Metacello-TestsMC

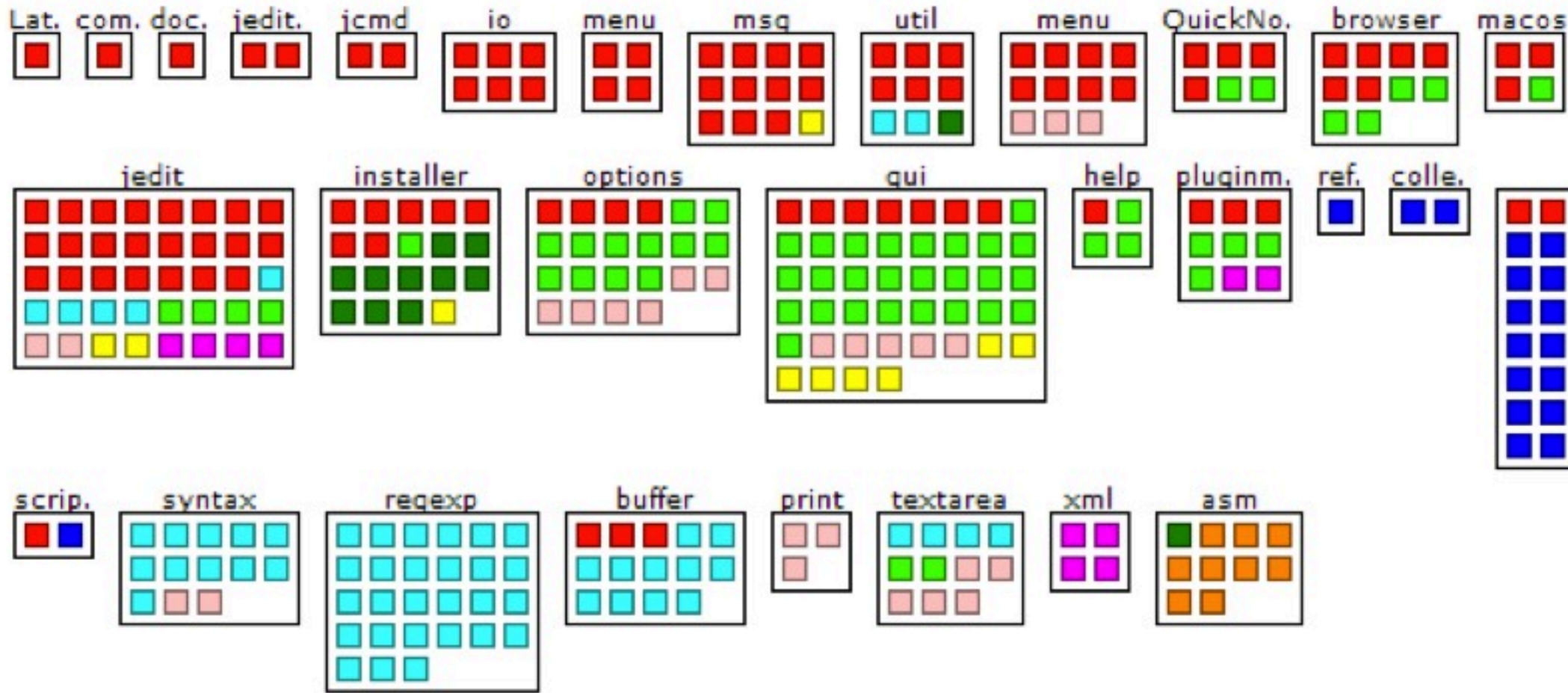


Metacello-TestsReference



Distribution map

Use to categorize existing entities into new groups or associates them with mutually exclusive properties



Lint

Set of rules to *statically* determine programming errors

Accessible from the refactor - code critic system browser menu

Code Critics: All checks on Mondrian (1857 problems)

Lint checks (1759)

Transformations (98)

Bugs
FAMIX
Fame
Grease

Grease Portability (214)

Intention revealing (192)

Miscellaneous (493)

Pharo bugs (1)

Possible bugs (178)

Seaside

Spelling (459)

Unnecessary code (222)

▲ #basicNew initialize is missing (52)

Empty Statements
Invalid object initialization

▲ Uses curly brace arrays (113)

Uses literal byte arrays

▲ Uses method annotations (41)

▲ Uses not portable class (8)

browse

refresh

Uses not portable class

Some classes are not portable across different Smalltalk dialects.

```
MOEvent>>#initializeEventon:  
MOEasel>>#initializeCompletionController  
MOAnnouncer>>#popupView:delay:zoomedInBy:  
MOAnnouncer>>#popupText:delay:  
MOAnnouncer>>#windowStatus:delay:  
MOAnnouncer>>#every.do:  
MOForceBasedLayout>>#doExecute:  
MORendererTest>>#TODOtestResize
```

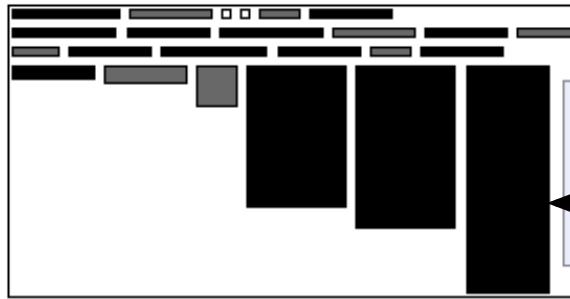
Profilers

Spy is a framework to easily build code execution profiler

CPU execution with Kai

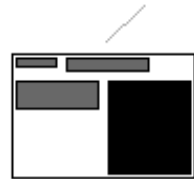
KaiProfiler

```
viewProfiling: [  
  | view |  
  view := MOViewRenderer new.  
  view  
    nodes: (1 to: 100)  
    forEach: [:each |  
      view nodes: (1 to: 100)].  
  view root applyLayout  
]
```



<MOGraphElement>applyLayout> 205.898 ms (53%)
 executed 10100 times
 invoked on at least 3732 receivers
 same return value per receiver

MOGraphElement



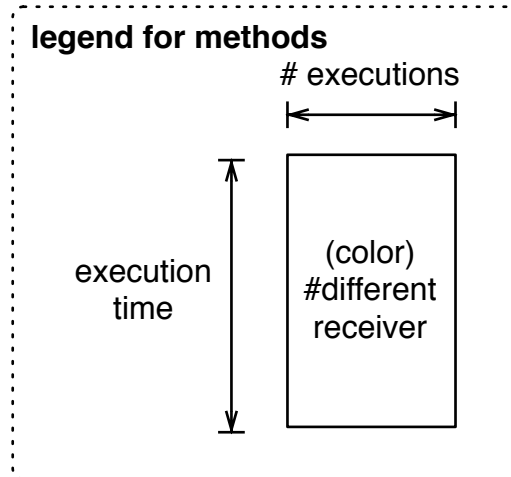
MOGraphElement

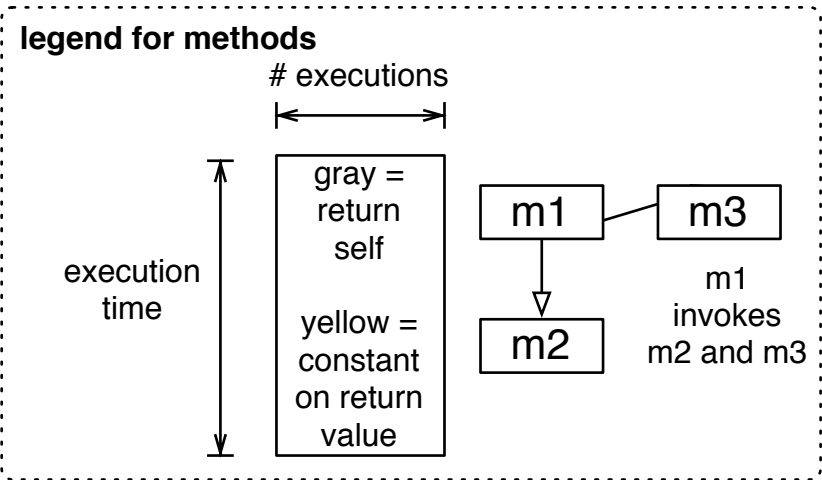
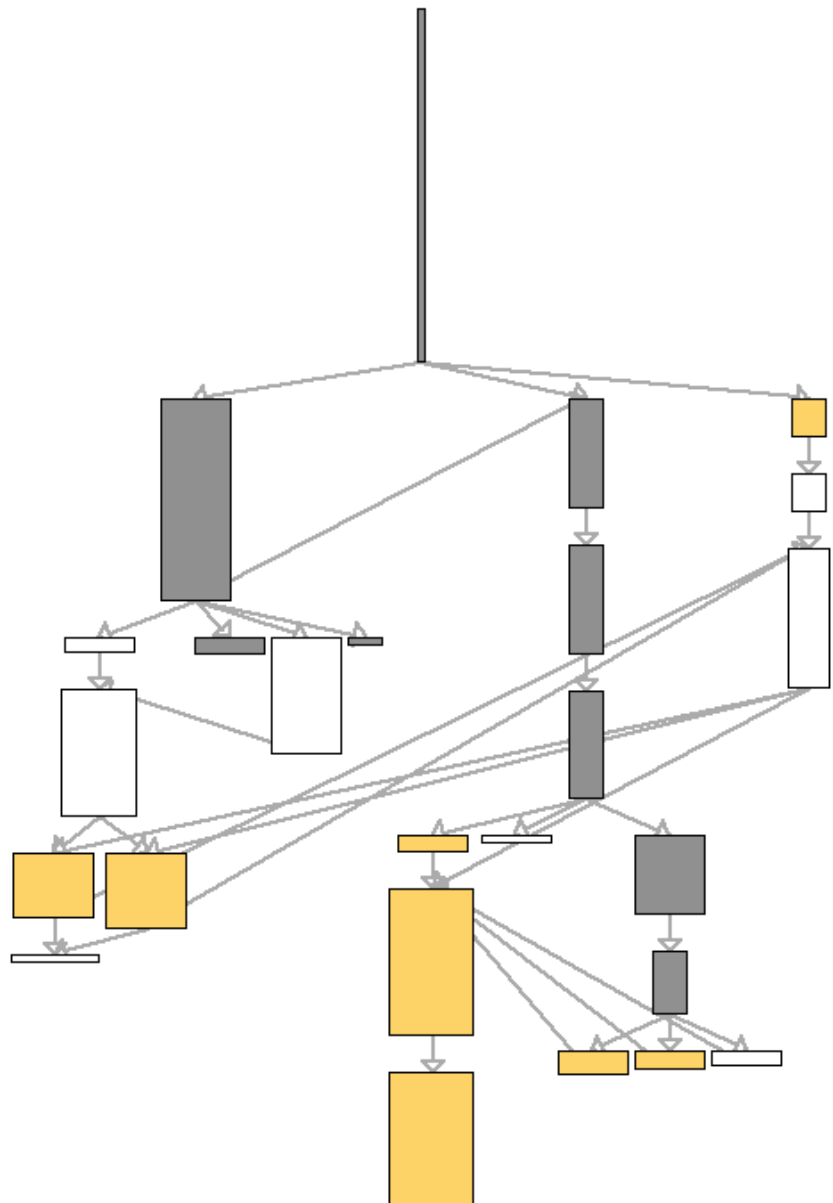


MOViewRenderer



MORoot

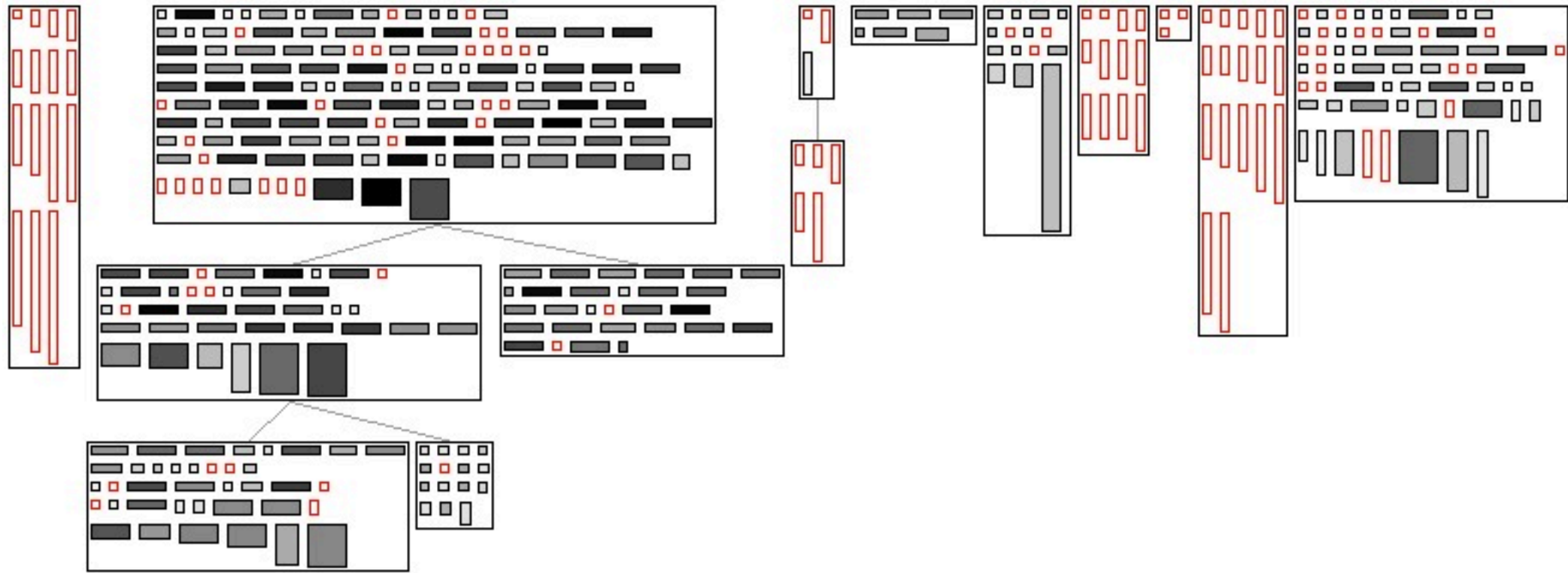




Test coverage with Hapao

HapaoTestCoverage

viewCoverageOfPackage: 'Mondrian'



height: number of lines of codes
width: number of execution (log scale)
color: number of different objects
red border color: not executed

software to analyze

XMLParser: David

Merlin: Vanessa

Metacello: Fabián

Monticello: Felipe

NewInspector: Roberto

Polymorph: Luis