Software Visualization Applied

S. Ducasse

rmod.lille.inria.fr / stephane.ducasse.free.fr

Merge
Trace
Reflection
Language
Validation
Semantic
Object
Analyses
Layers
Traits
Architecture
Evolution
Aspects
analysis
Serialization
Bootstrap
Teaching
Engineering
Information
Reflective
Tools
Pharo
Visual
Dependency
Repositories
Code
Encapsulation
Understanding
Identification
Moose
Quality
Reengineering
IDE
Software
Comprehension
Class
Scripting
Models
Programming
History
Dynamic
Maps
Development
Meta
Modules
Interexchange
Proxy
Duplicated
Remodularisation
formats
Checking
Reverse
Visualization
Mining
Metrics
Array
Web
Extraction
Rules
Extensions
Design
Patterns
Feature
Cycles

Working on more than program visualization...

Code analysis

Software metrics

Quality model

Dynamic analysis

Refactorings

Software remodularisation

Cycle and layer identification

Tool building

Rules

Changes characterisation

Architecture extraction

Mining software repositories

Support for Merging

Traits: orthogonal inheritance

Modules

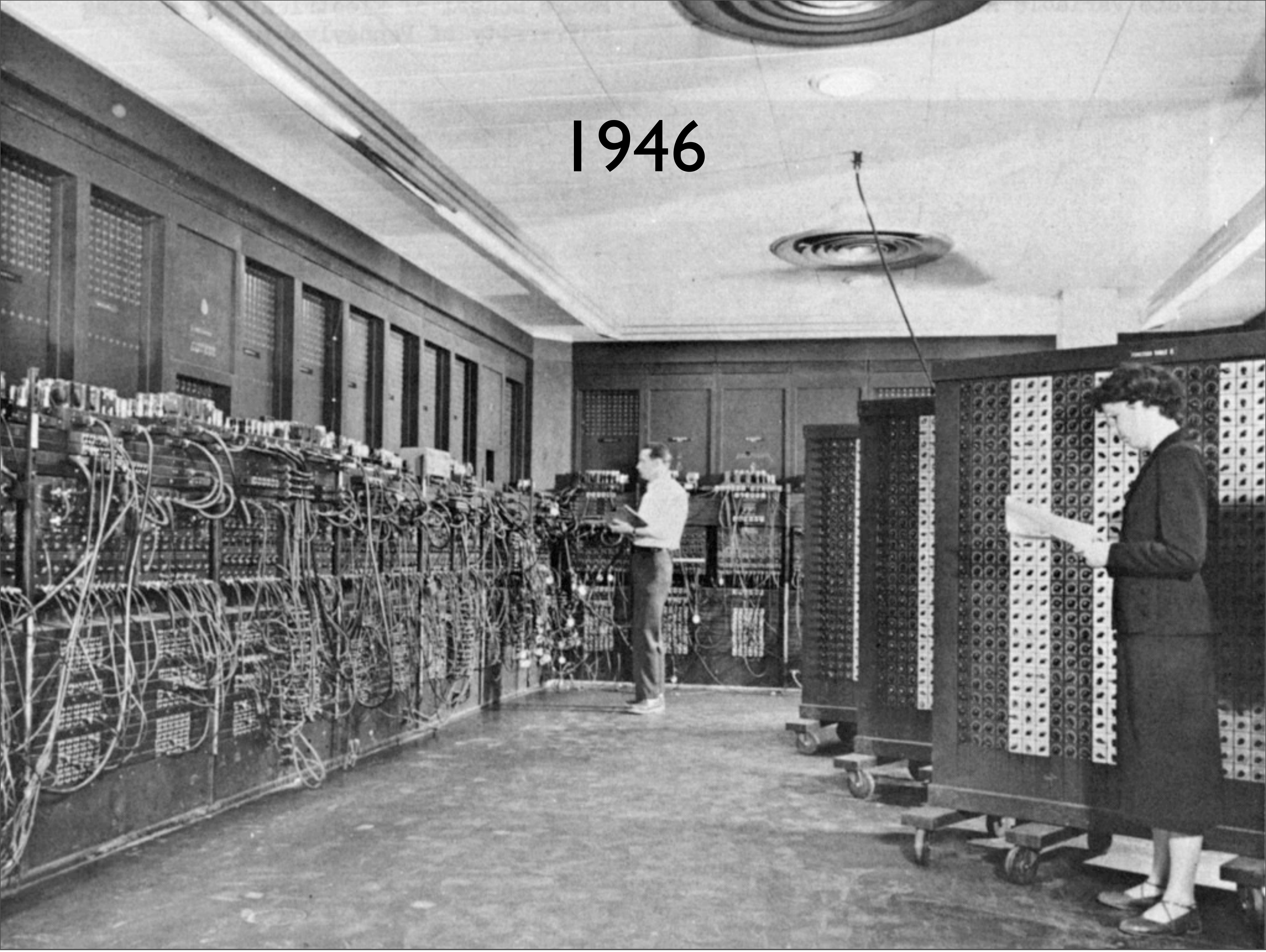
Class extension scoping

Program isolation

Component model

Reflective programming

RMod



1946

How large is your project?

1’000’000 lines of code

* 2 = 2’000’000 seconds

/ 3600 = 560 hours

/ 8 = 70 days

/ 20 = 3 months

Where are located the classes containing most of the bugs?

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

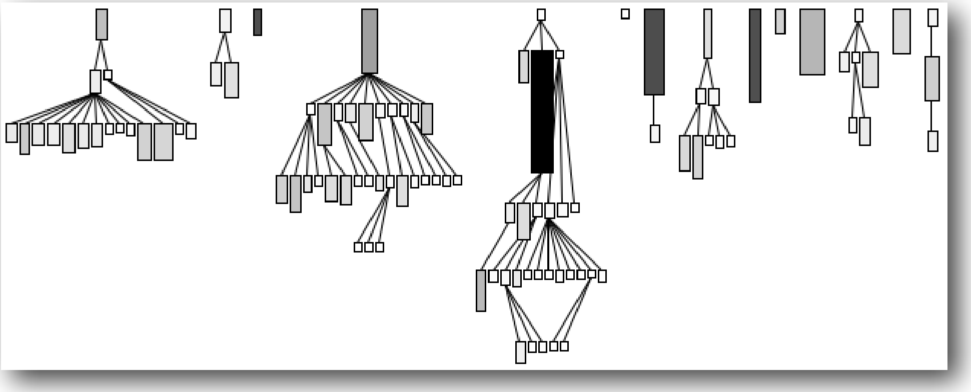
Limited short-term memory (three to nine)

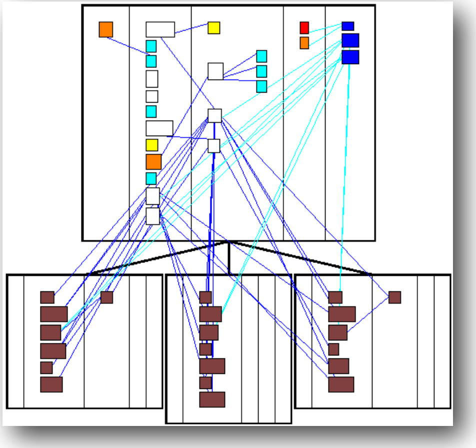
Difficult to remember too many symbols/semantics

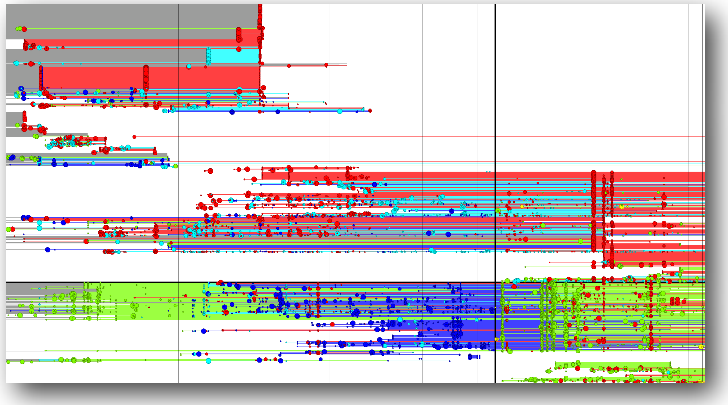
Culture, Colorblind

Visualization principles in 3 min

- Preattentive visualization (unconscious < 200ms)
- Gestalt principles (from 1912)
- 70% of our sensors are dedicated to vision







How many 5?

3332123466509000096766689877835367
7866760910919818971746433039821768
34467865860880221167687687789762

How many 5?

3332123466**5**0900009676668987783**5**367
7866760910919818971746433039821768
3446786**5**860880221167687687789762

Preattentive attributes

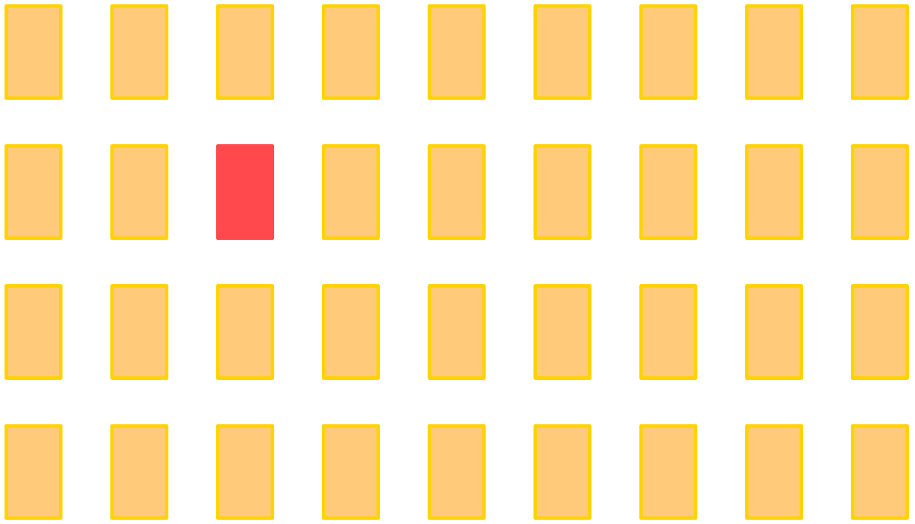
Color intensity

Form: orientation, line length, line width, size, shape, added marks, enclosure

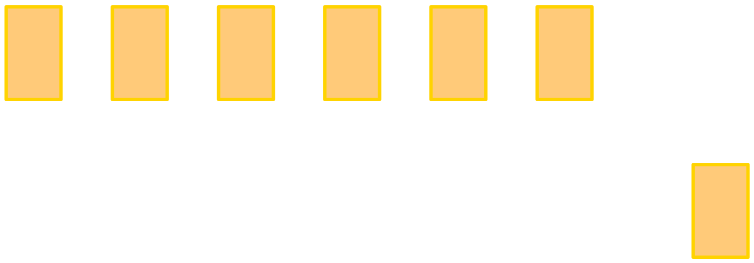
Spatial position (2D location)

Motion (flicker)

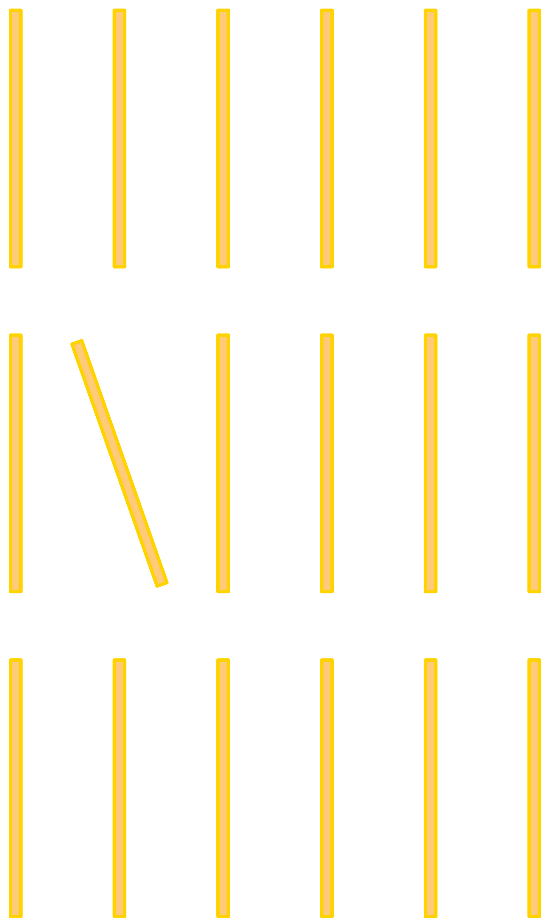
Color / intensity



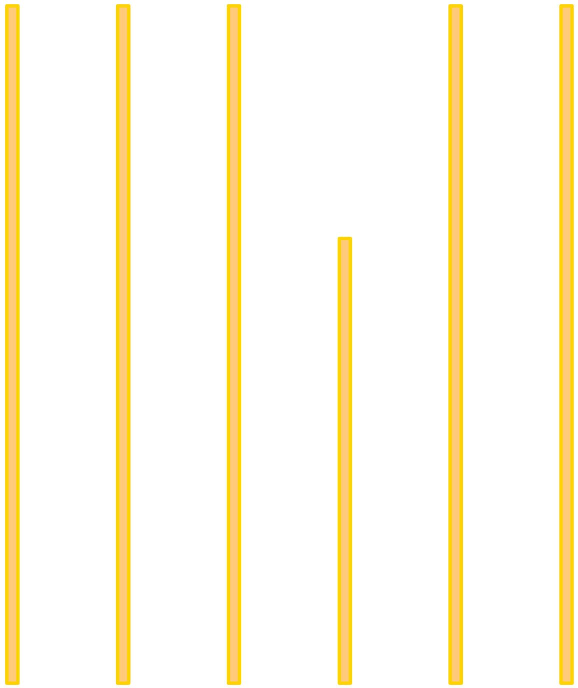
Position



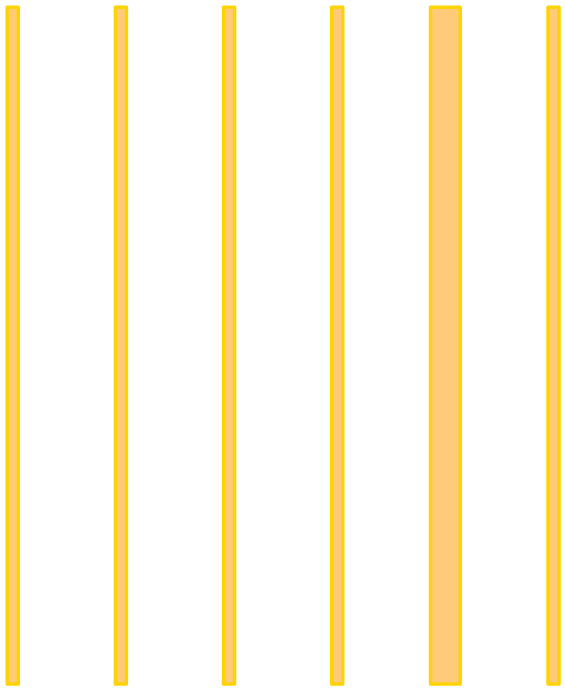
Form / Orientation



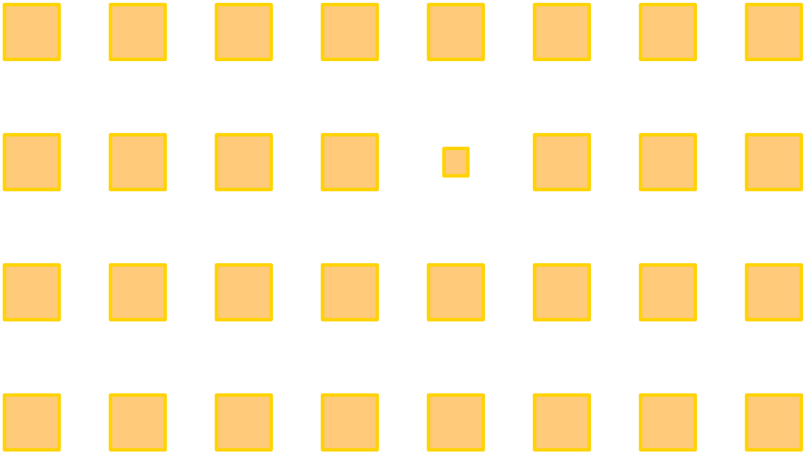
Form / Line length



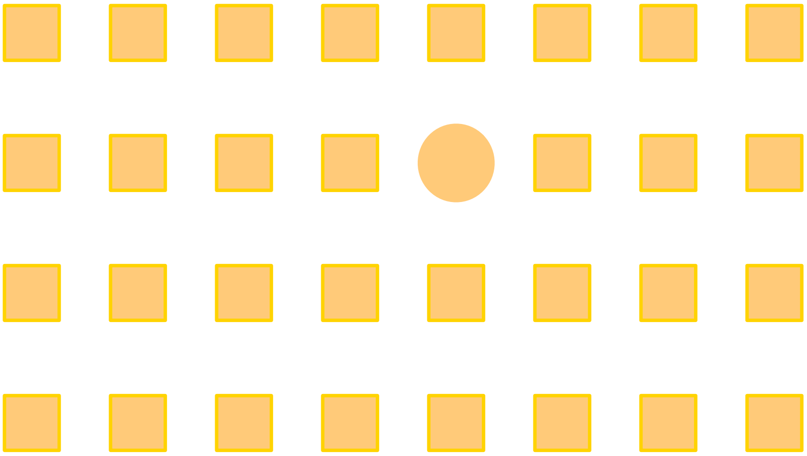
Form / Line width



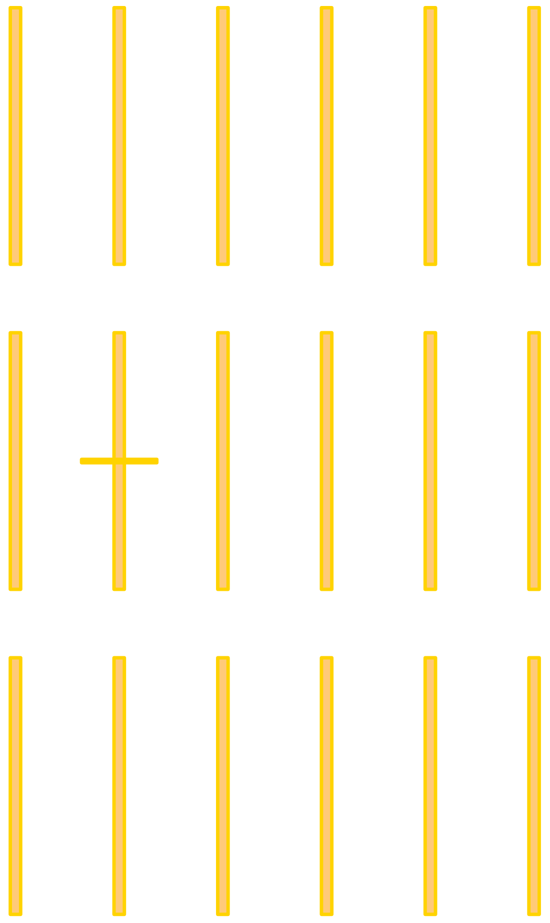
Form / Size



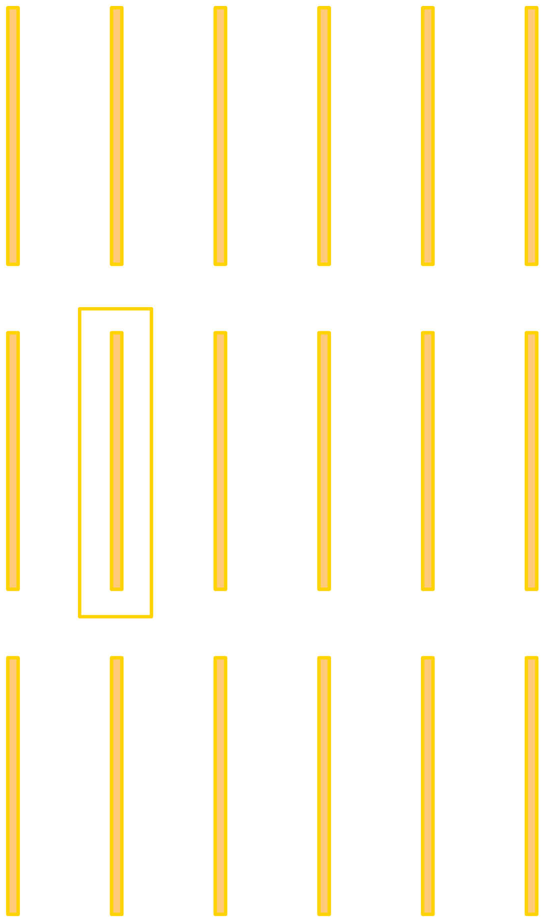
Form / Shapes



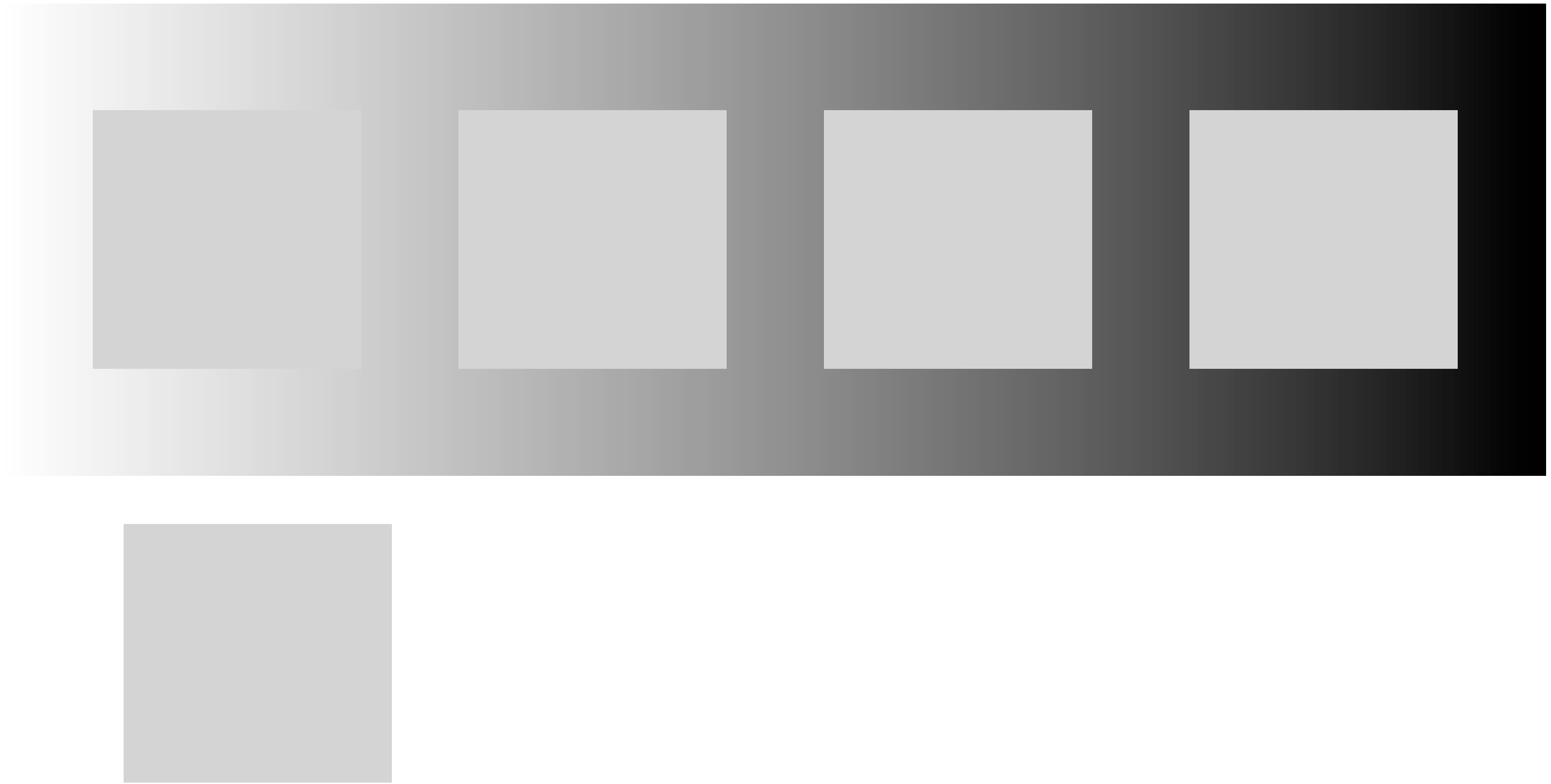
Form / Added marks



Form / Enclosure



Context



Gestalt Principles of Visual Perception

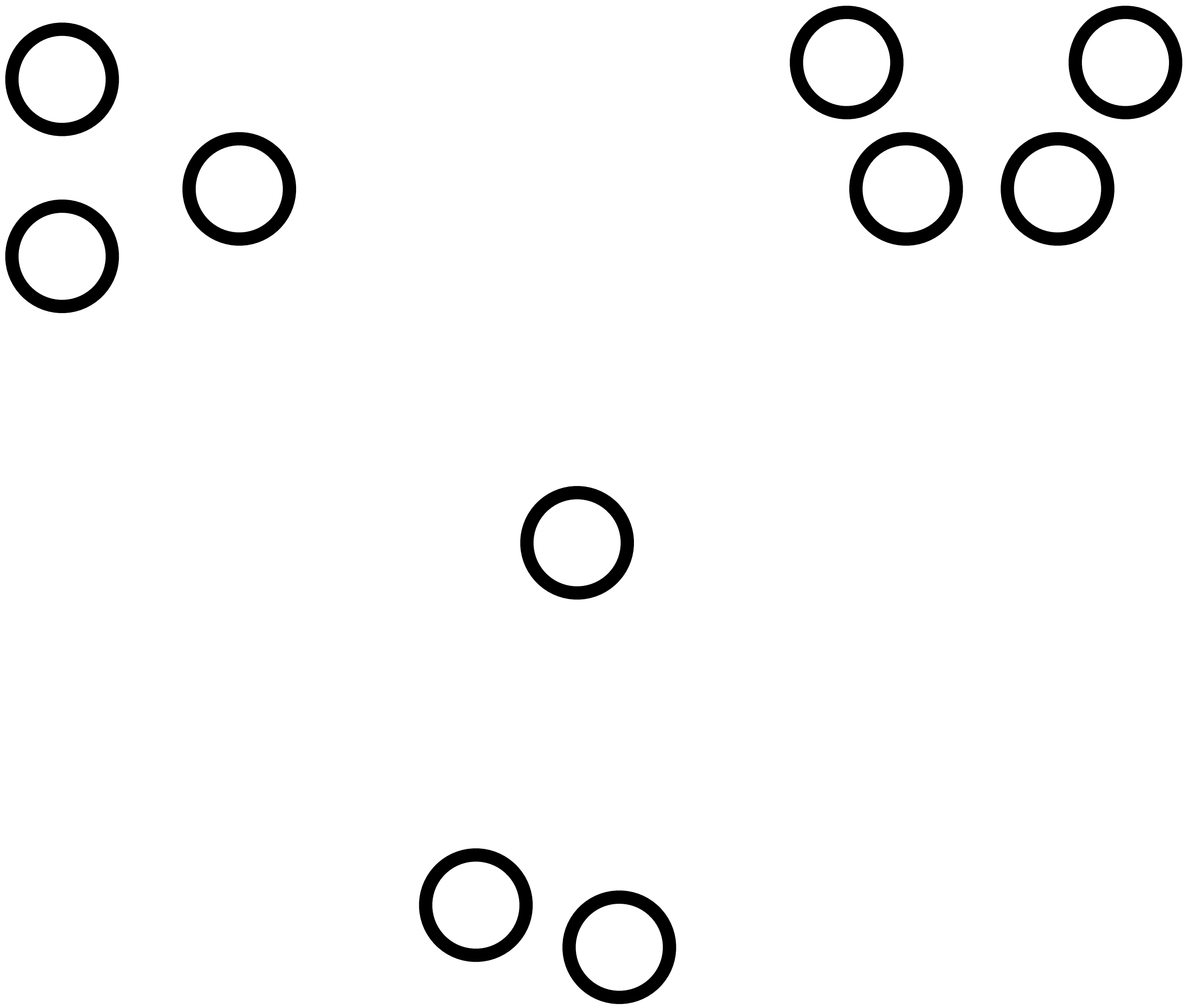
Back in 1912, from the Gestalt School of psychology

Still stand today

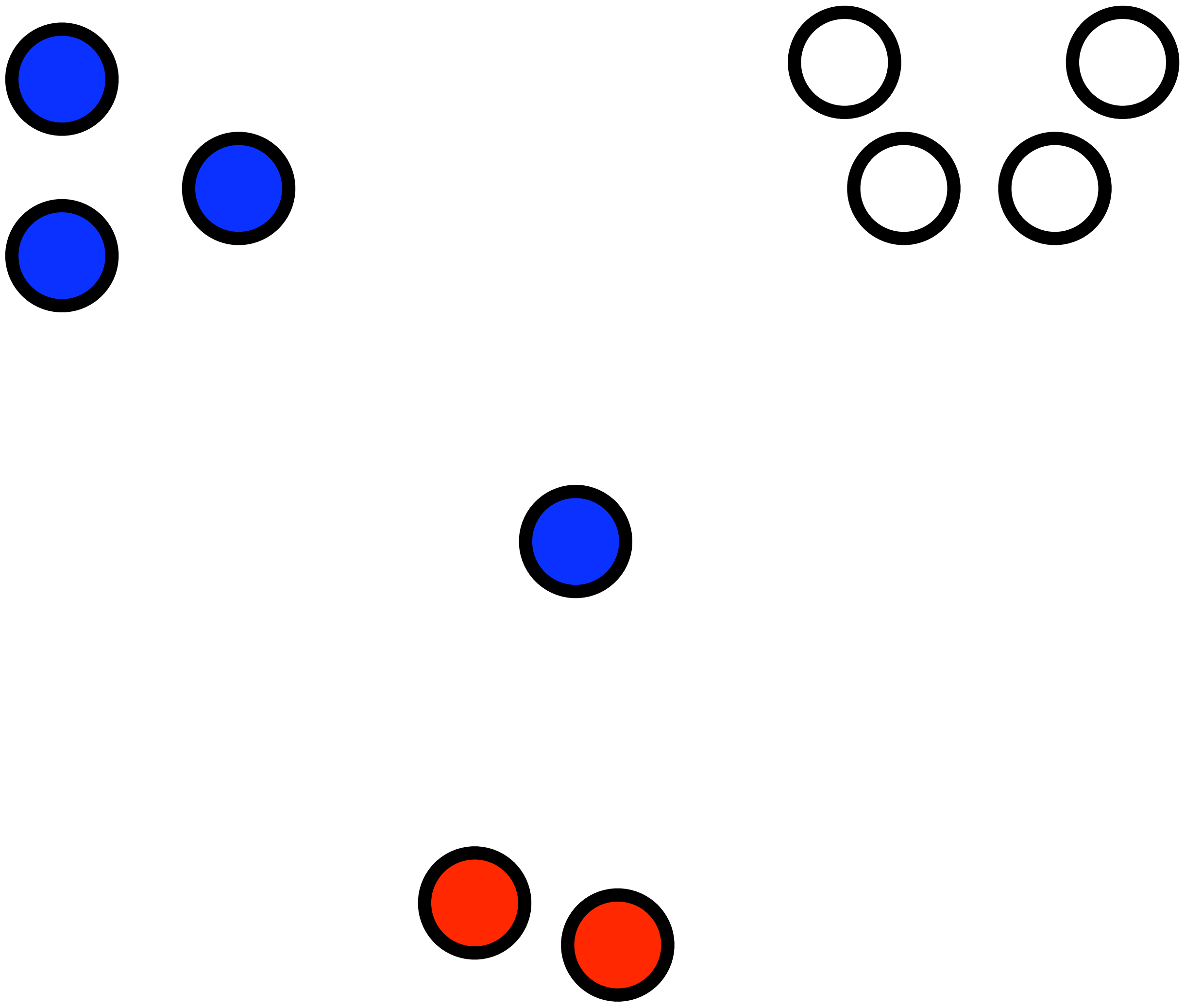
Gestalt means patterns

How do we perceive pattern, form, and organization?

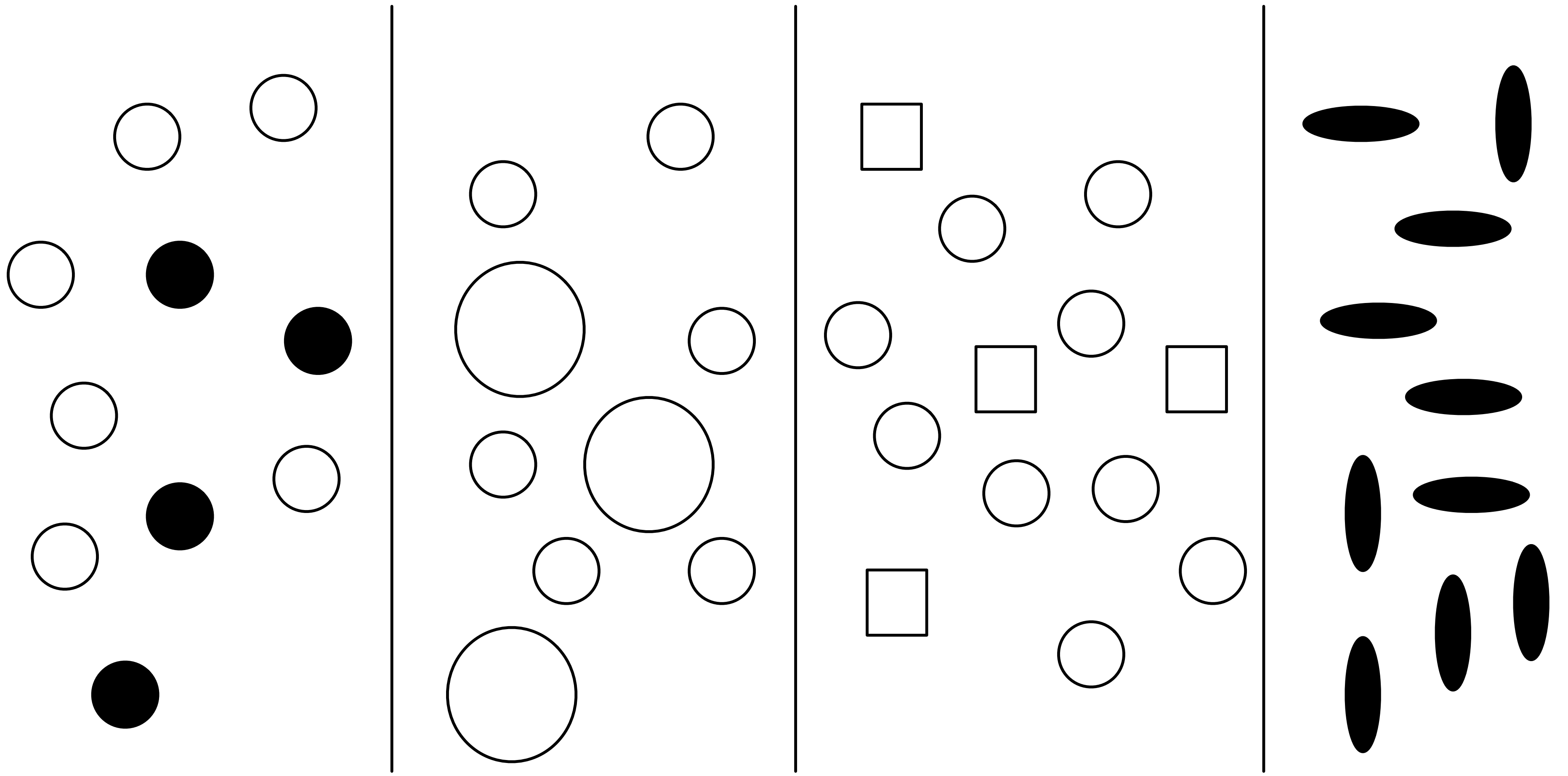
Principle of Proximity



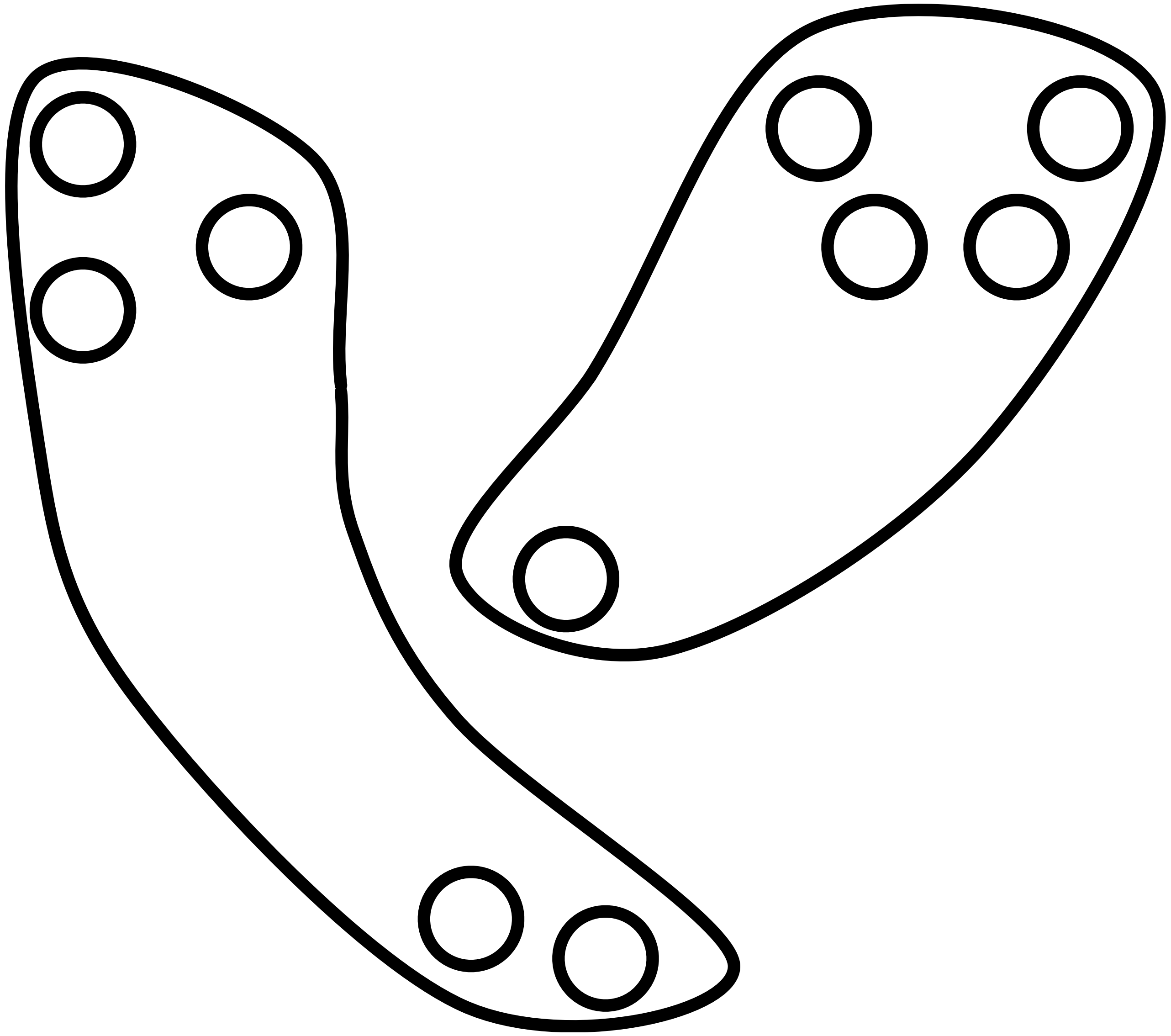
Principle of Similarity



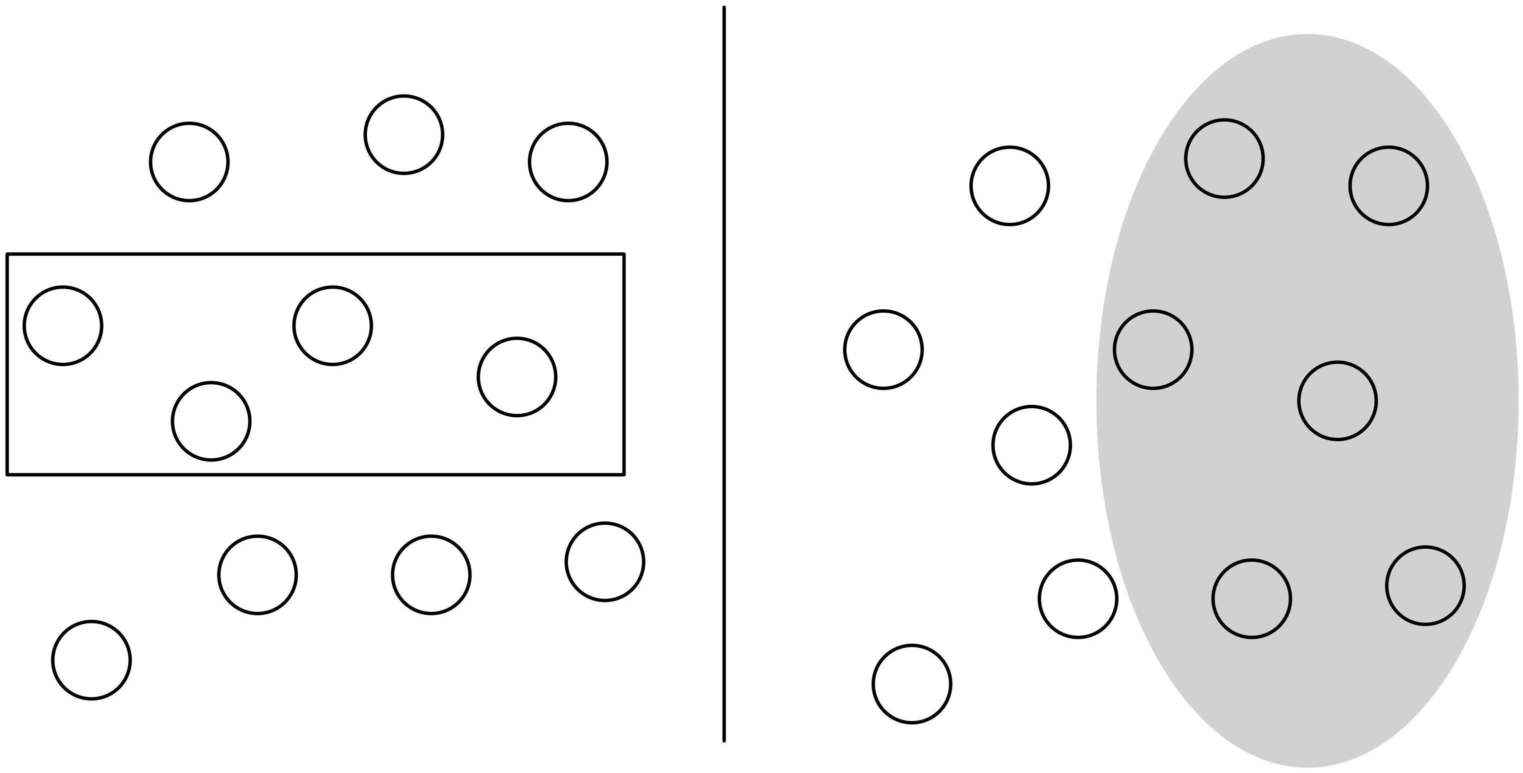
Principle of Similarity



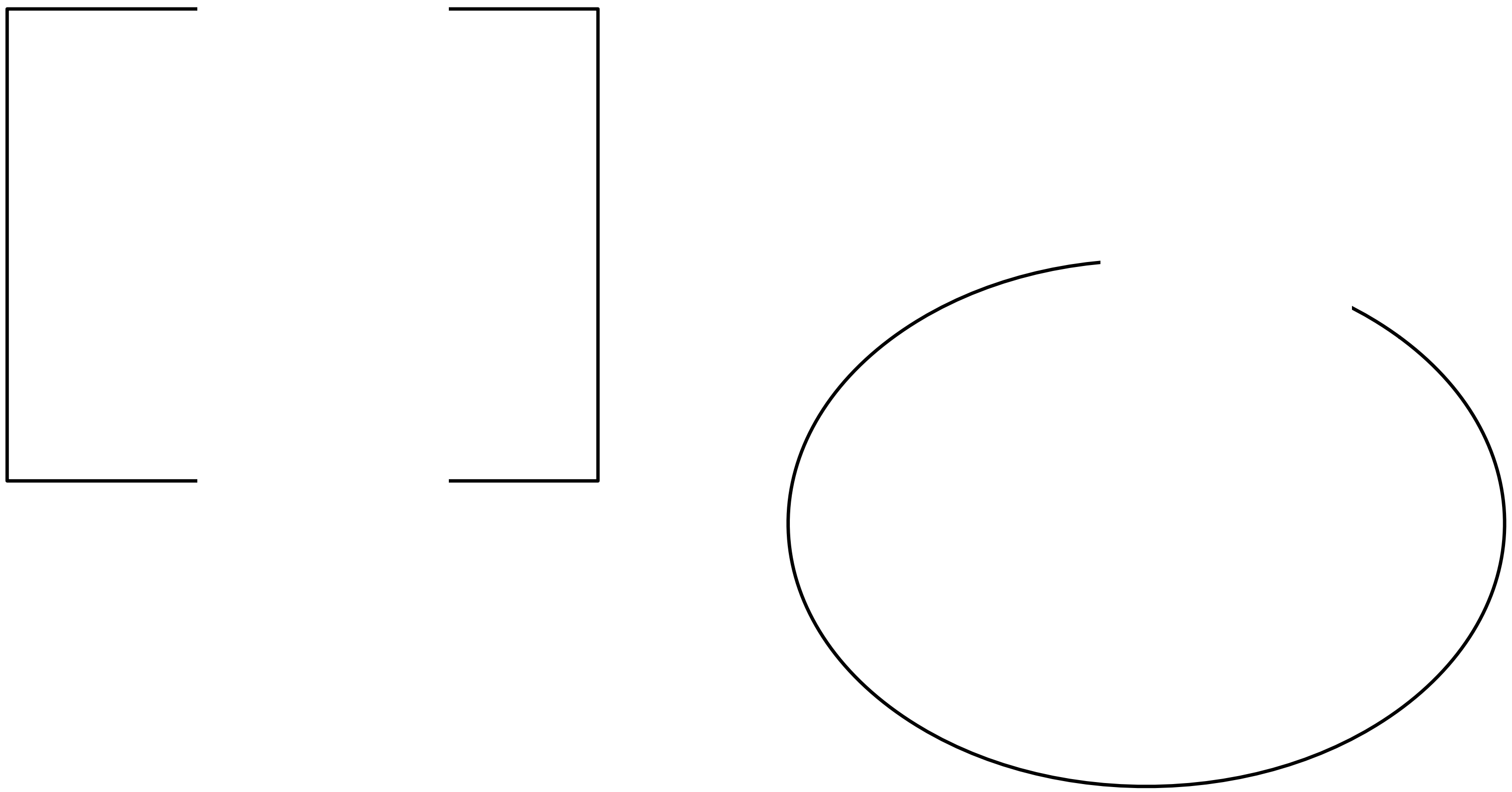
Principle of Enclosure



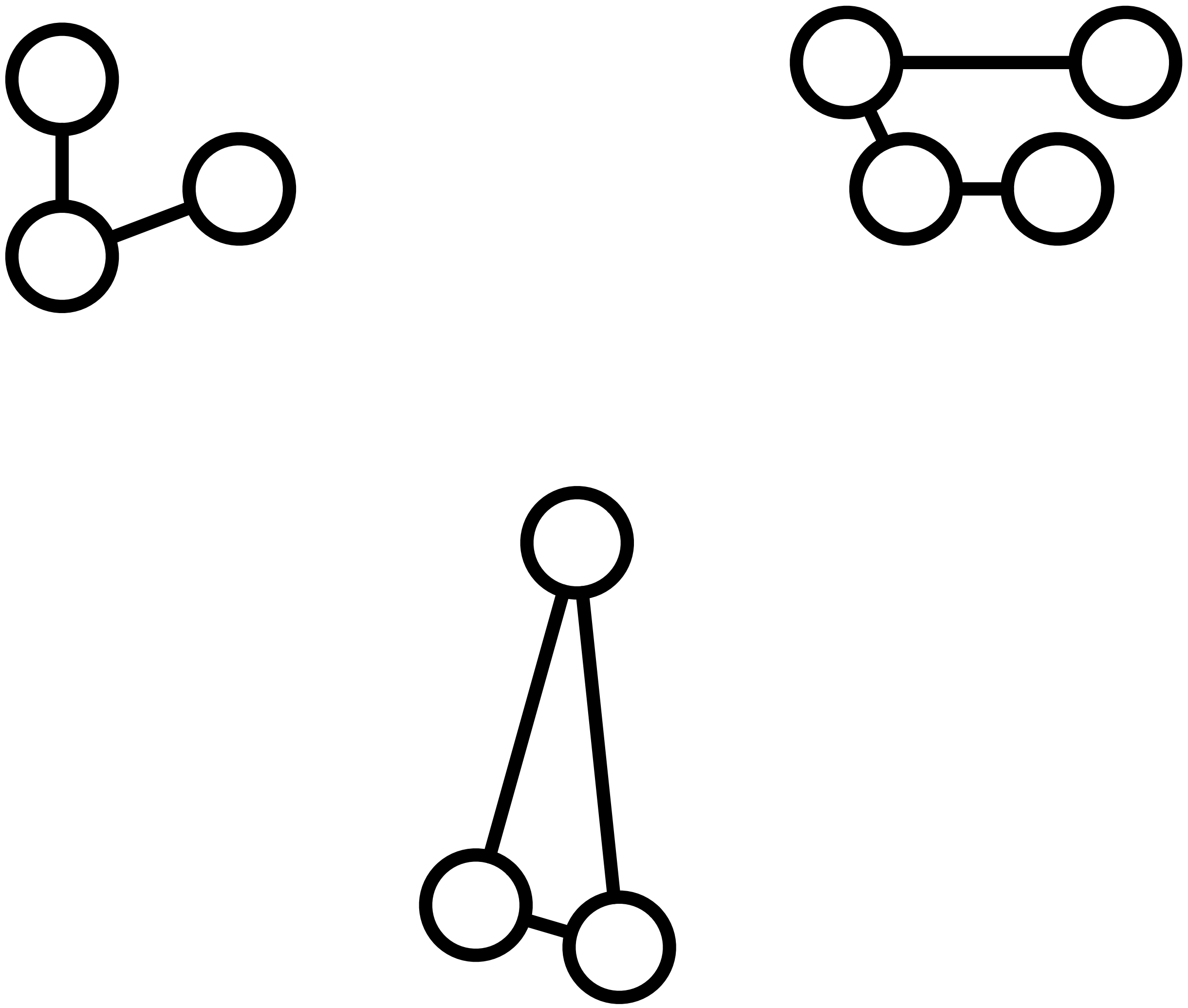
Principle of Enclosure



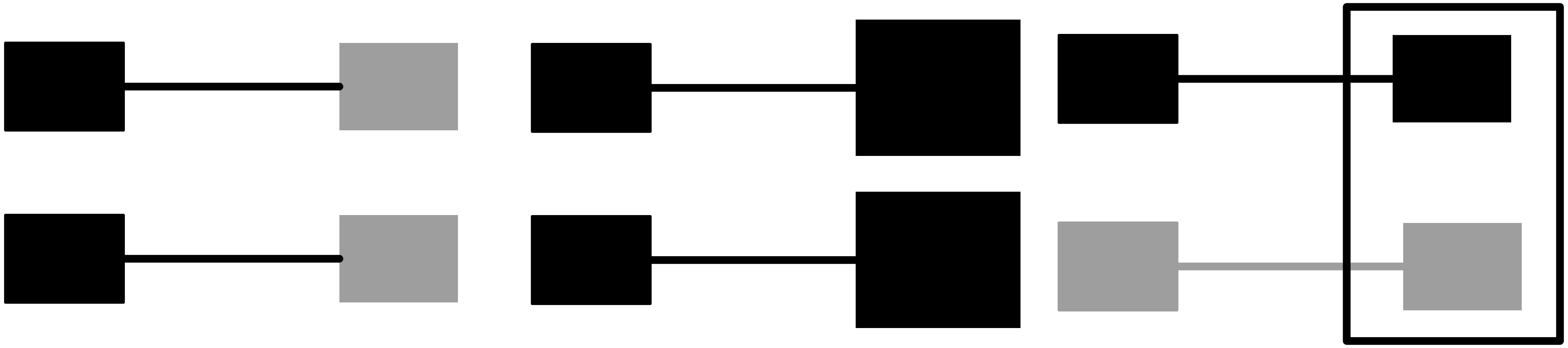
Principle of Closure



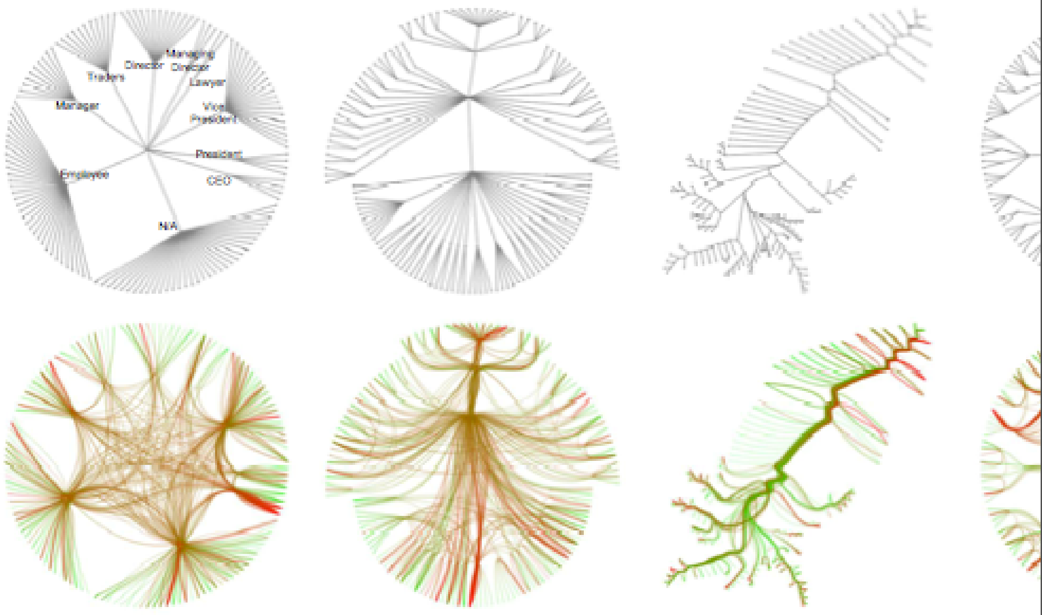
Principle of connectivity



Principle of connectivity



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

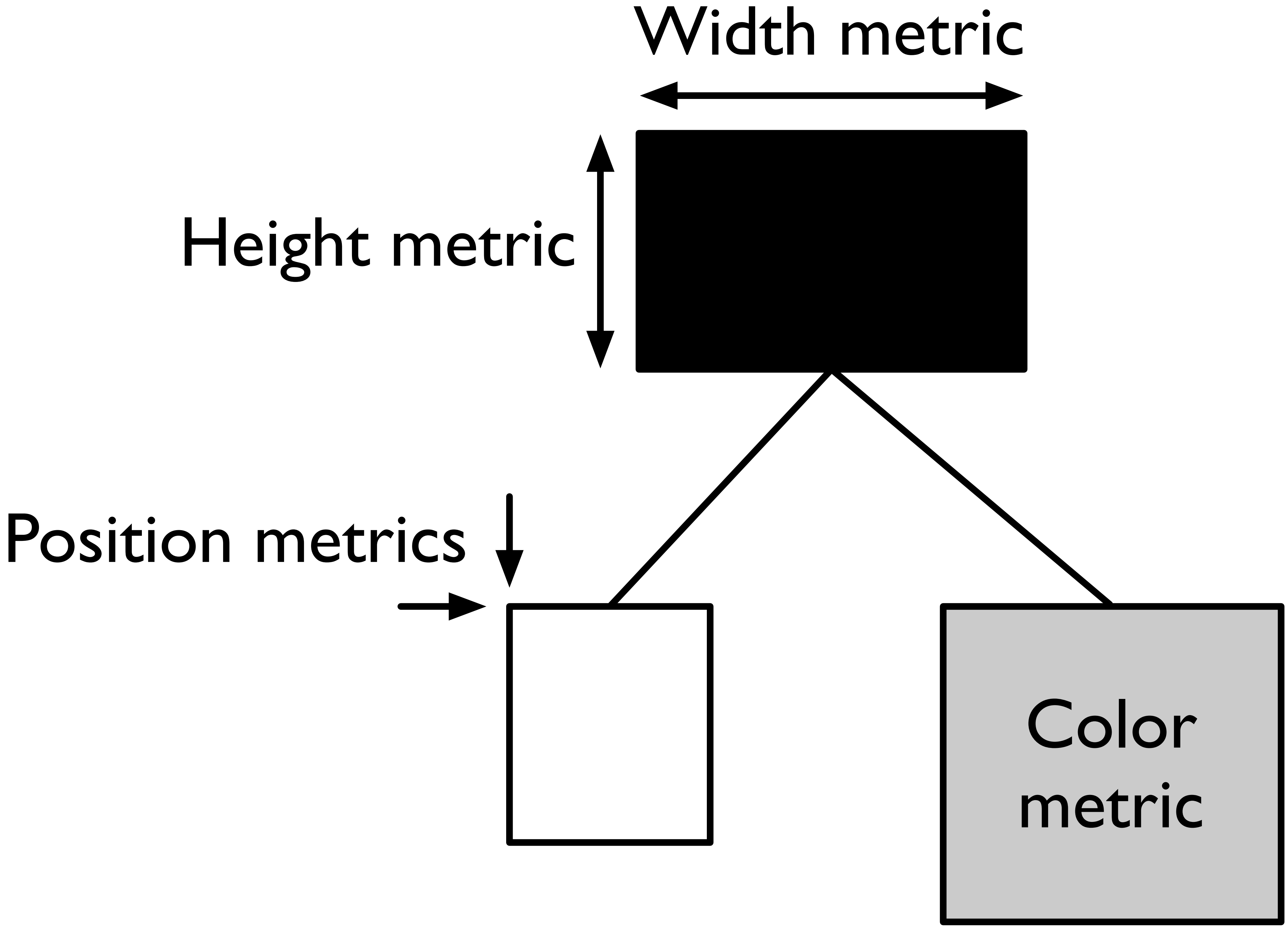
- Understanding code is difficult!
- Systems are large
- Code is abstract
- Should I really convinced you?

- Some existing approaches
  -Metrics: you often get meaningless results once combined
  -Visualization: often beautiful but with little meaning

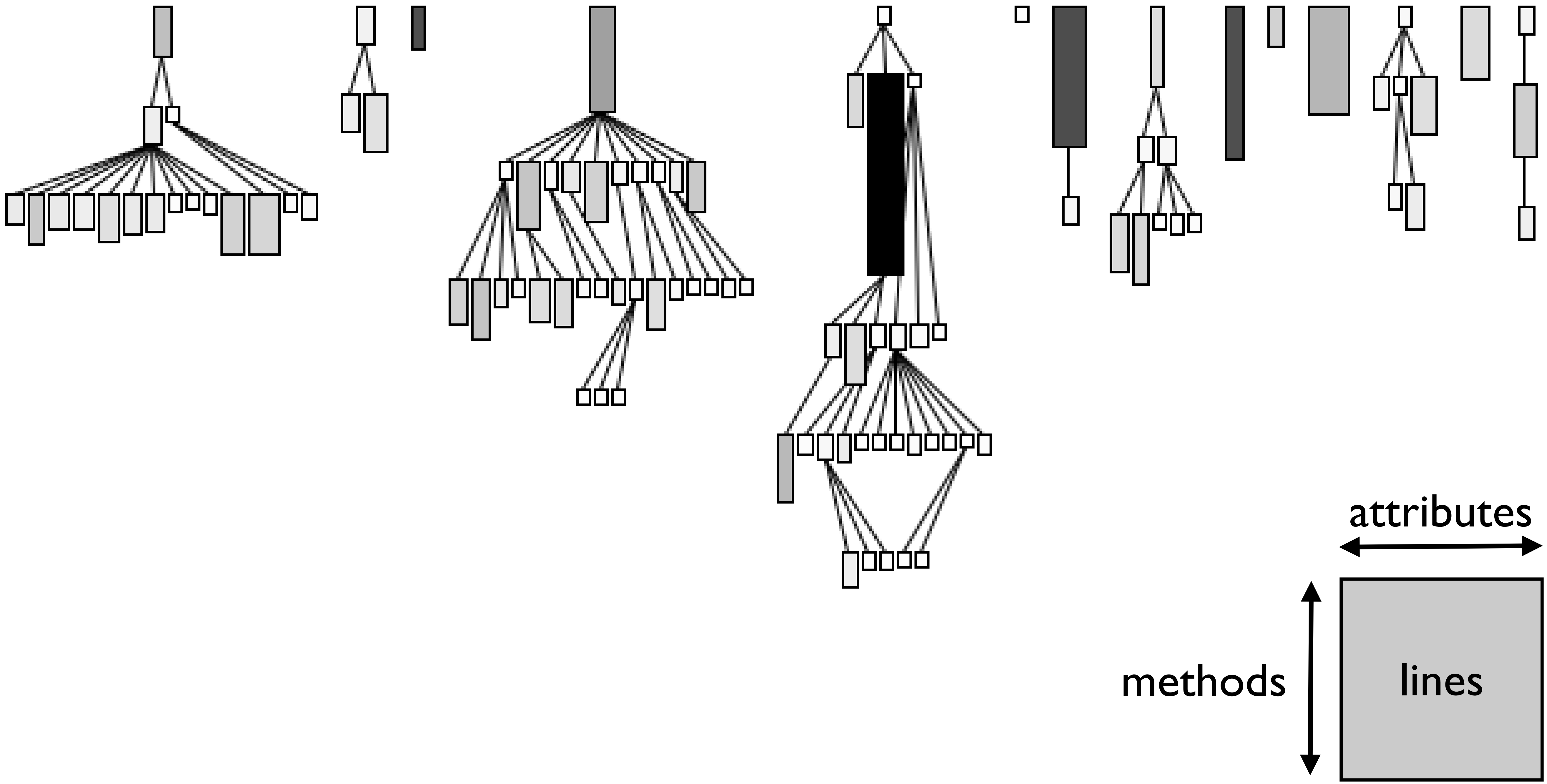
- Polymetric view is an idea of M. Lanza [WCRE,TSE]

Polymetric views show up to 5 metrics.

Lanza etal, 03

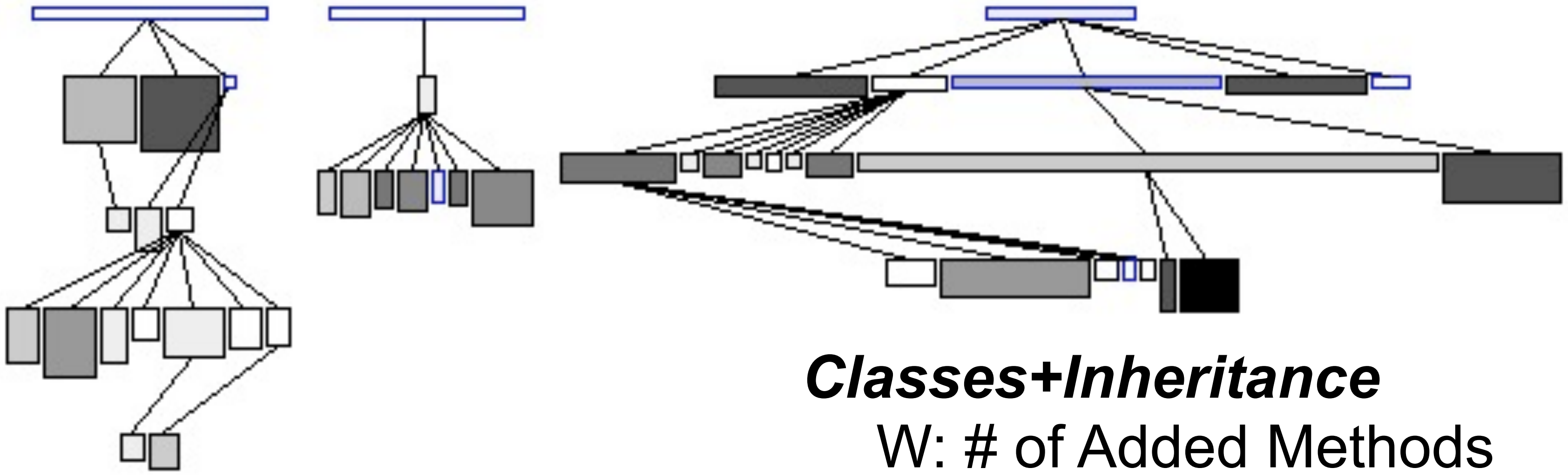


System Complexity shows class hierarchies.



Polymetric views condense information

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

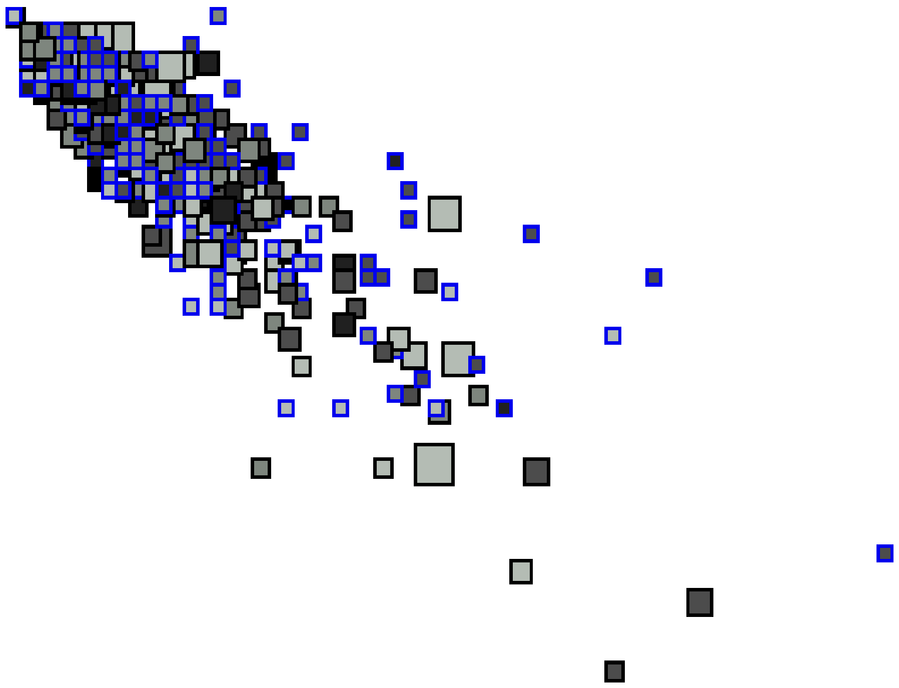


Classes+Inheritance

W: # of Added Methods

H: # of Overridden Methods

C: # of Method Extended



methods

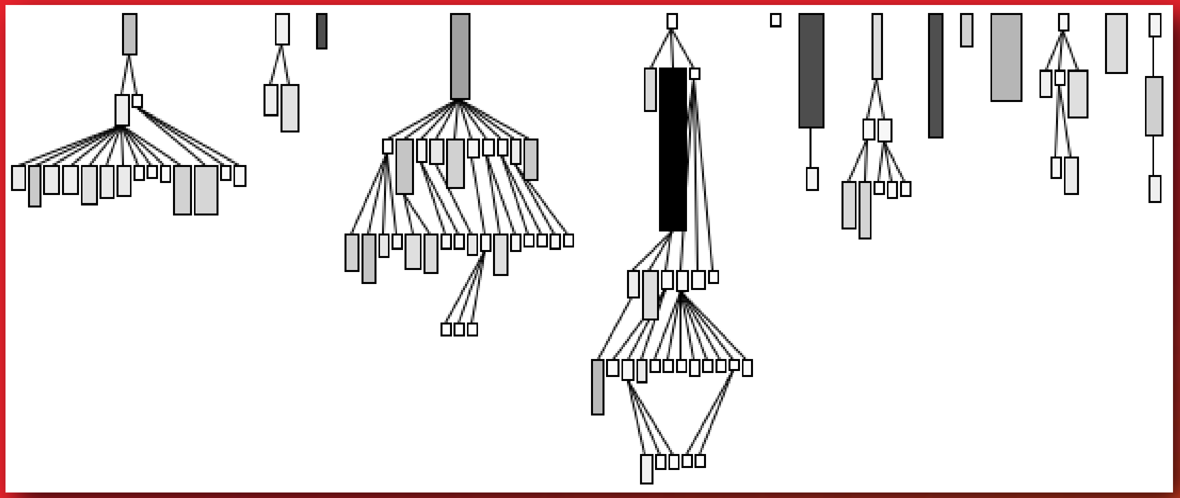
LOC

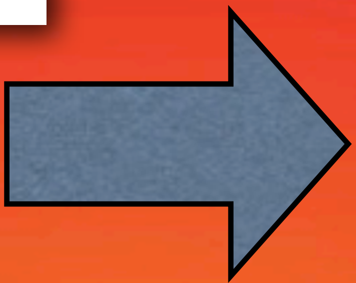
# statements

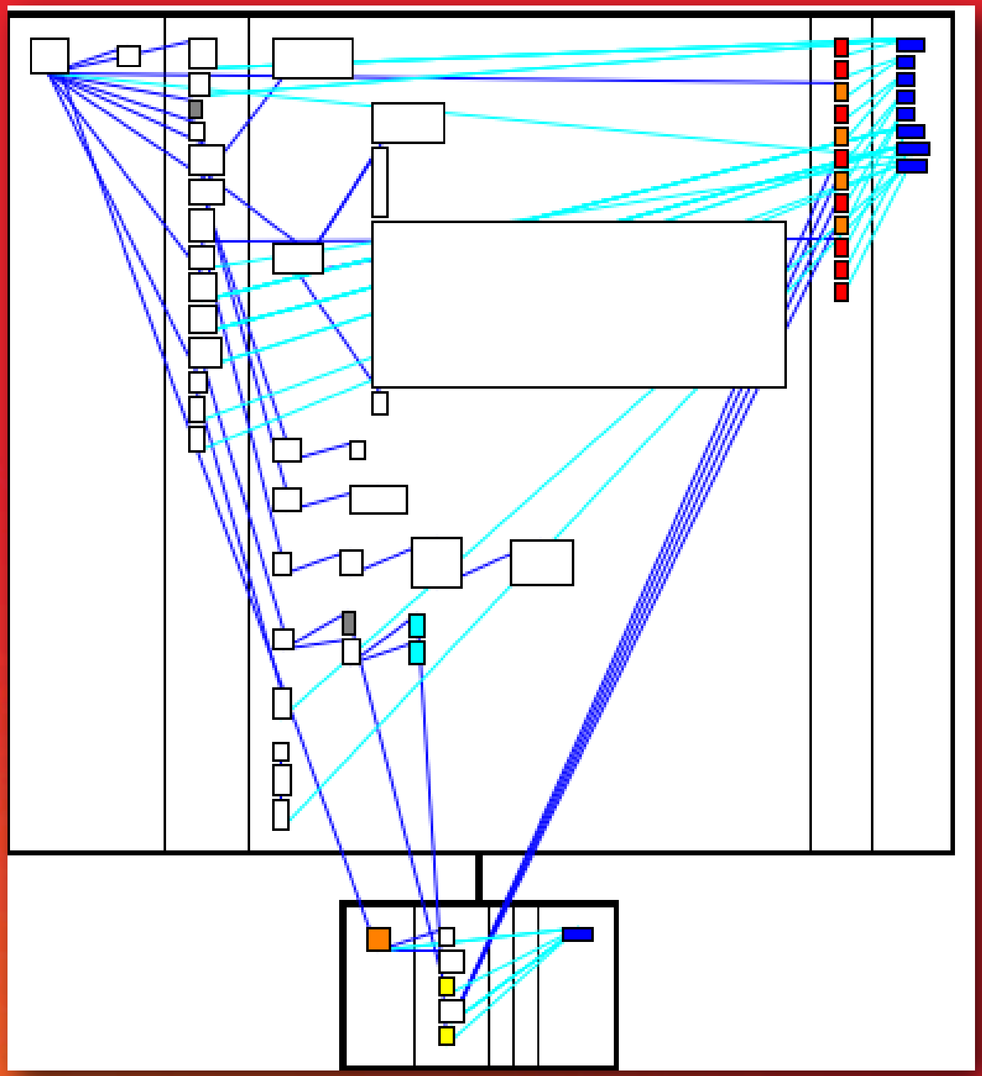
# parameters

Understanding Classes: Easier?

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



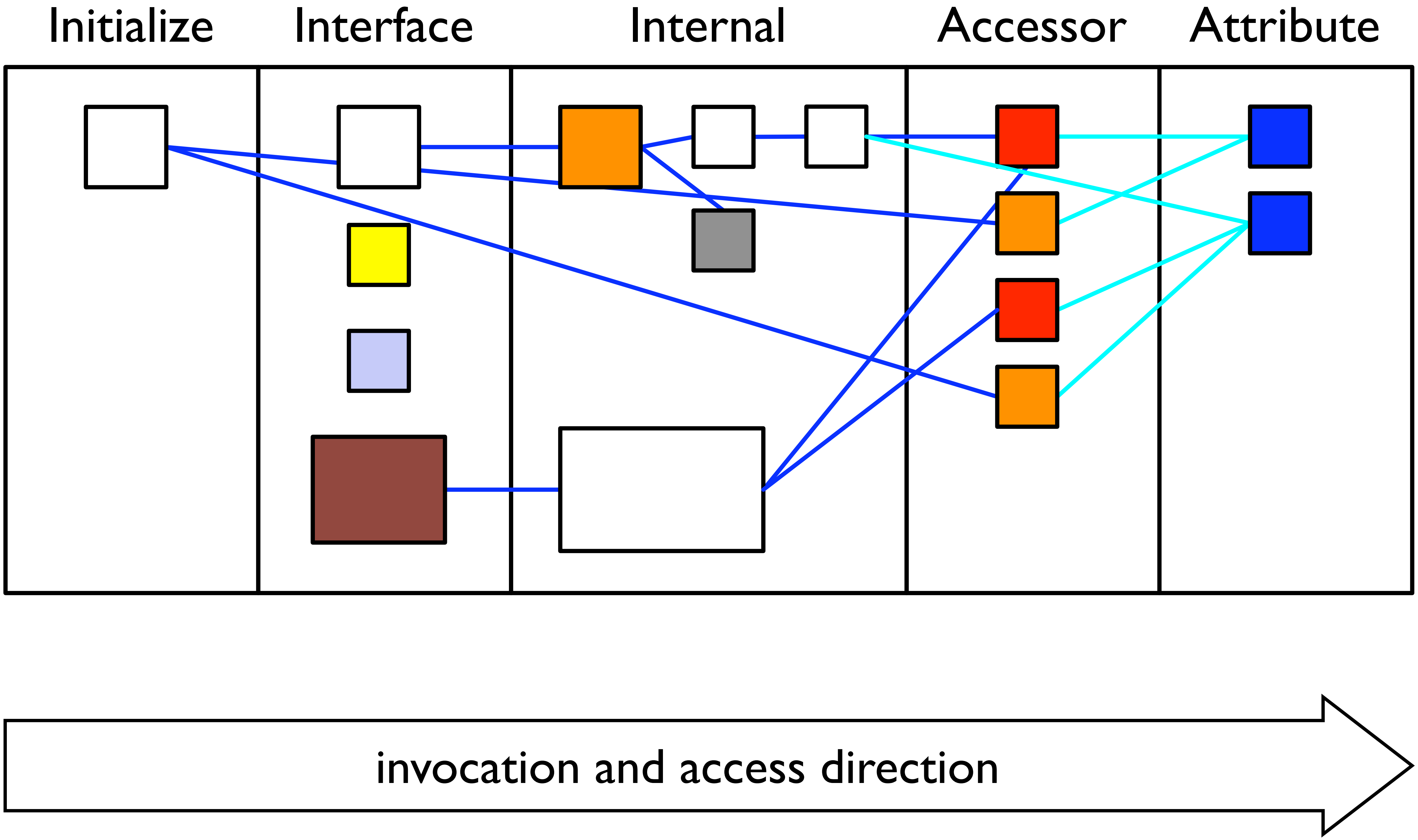




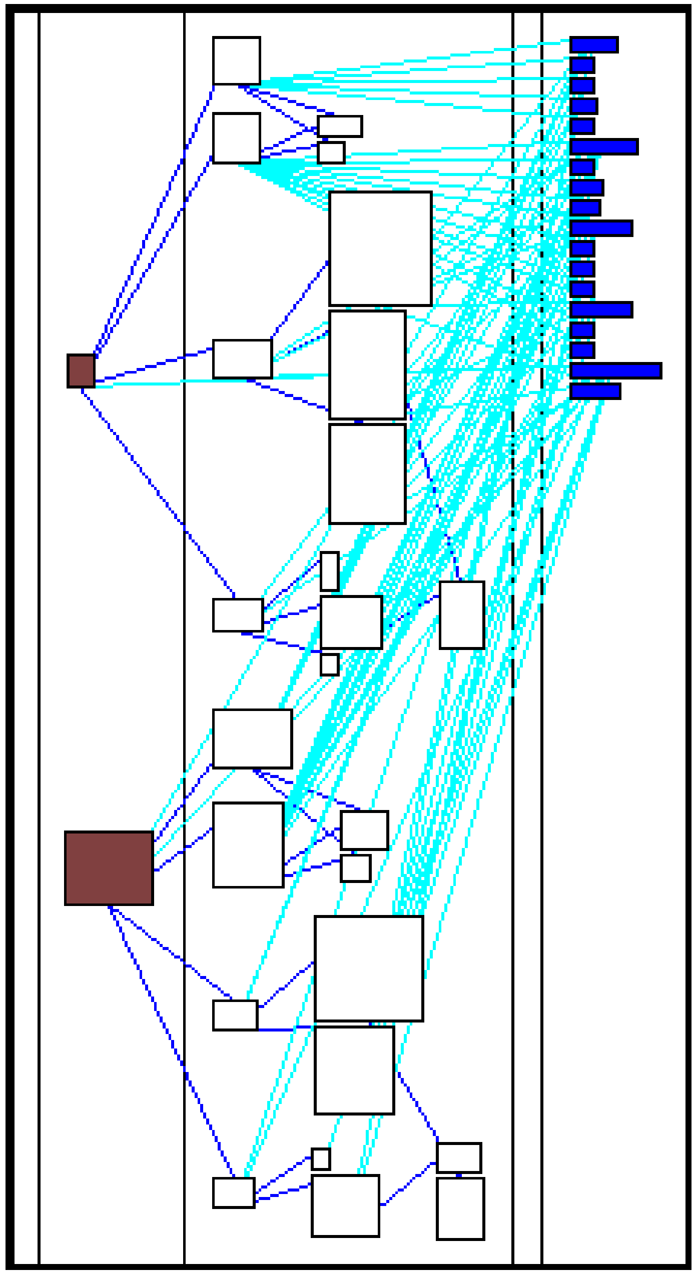
- Class blueprint is an idea of M. Lanza [OOPSLA]

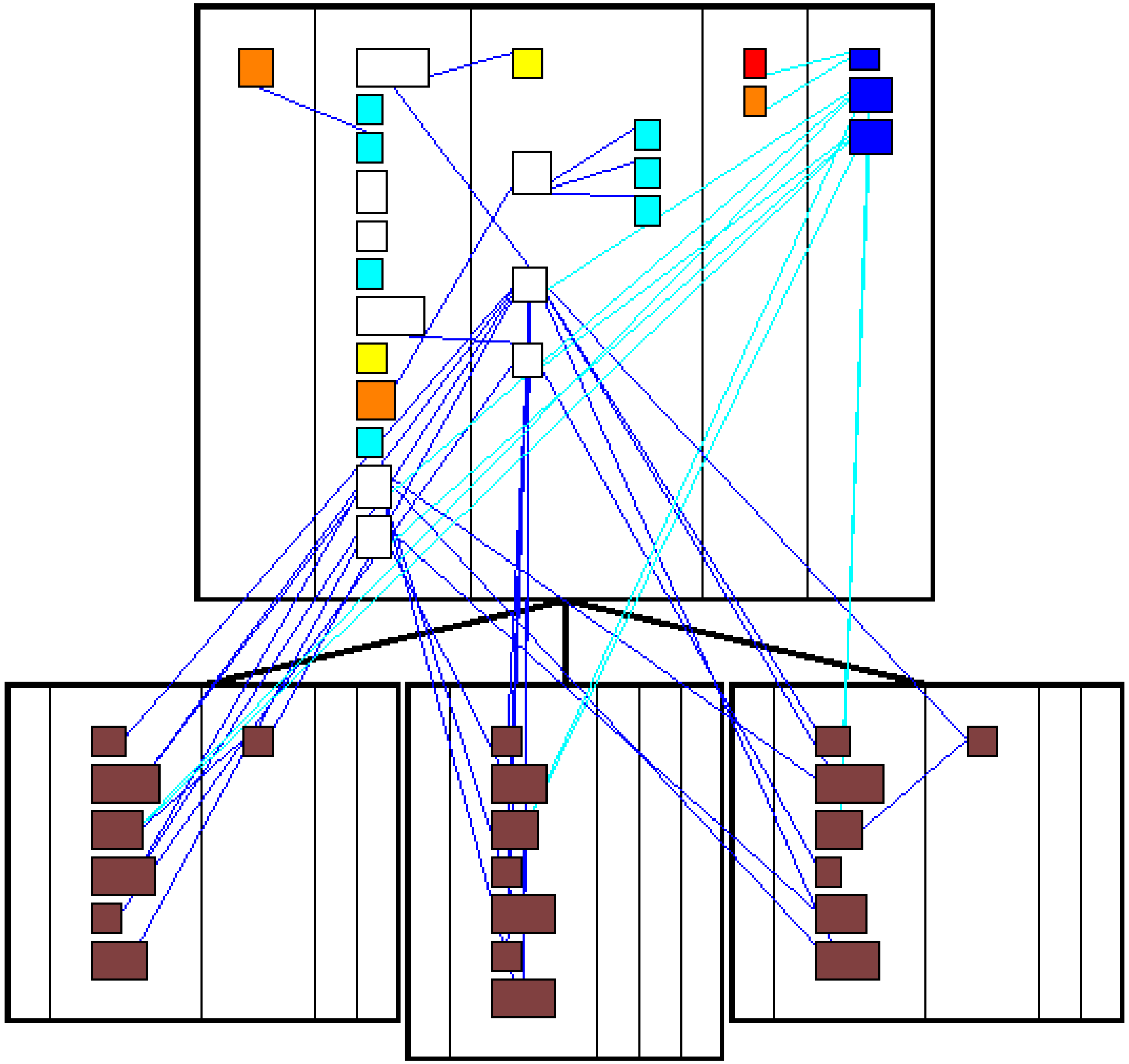
Class Blueprint shows class internals.

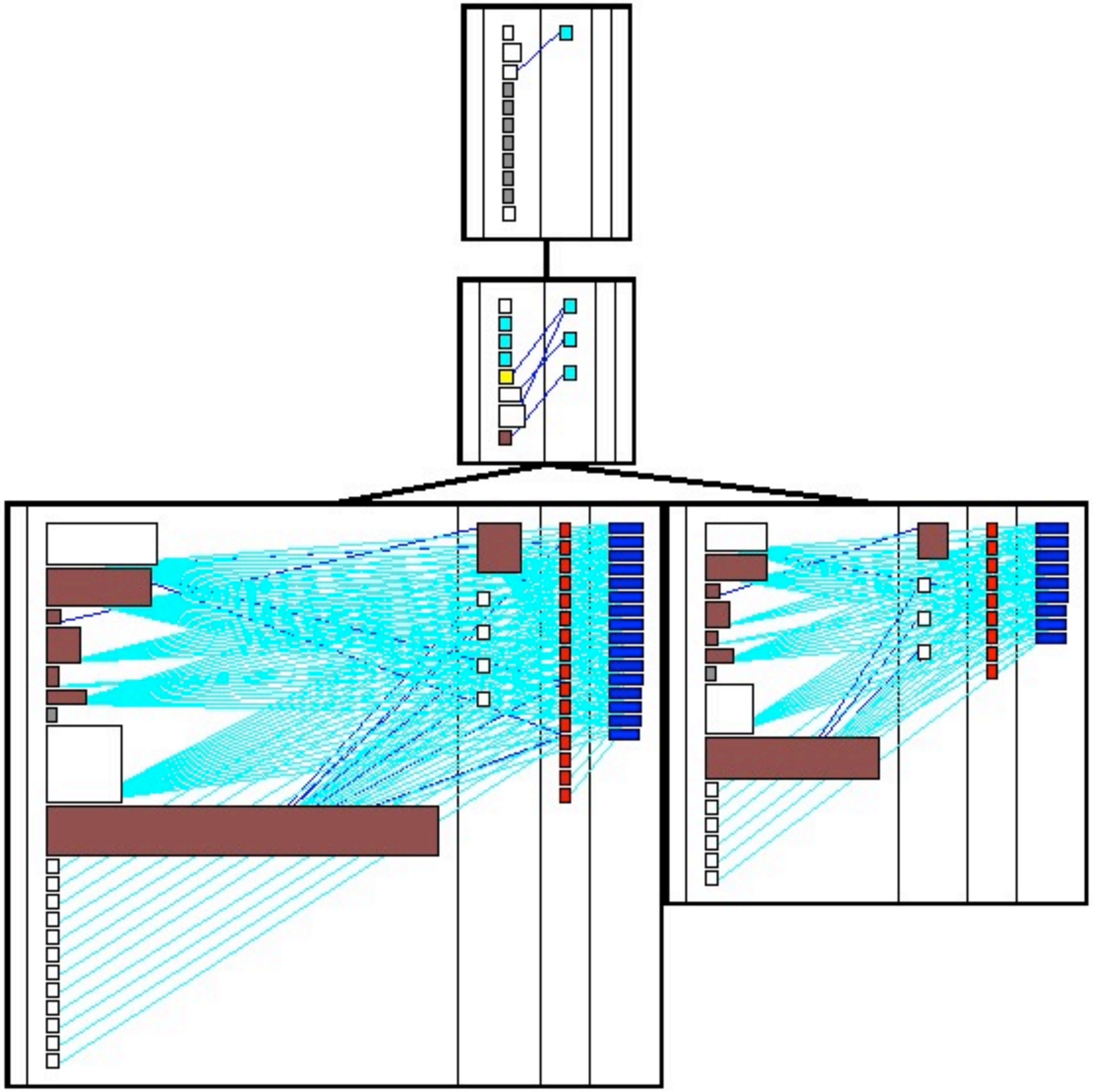
Ducasse, Lanza, 05

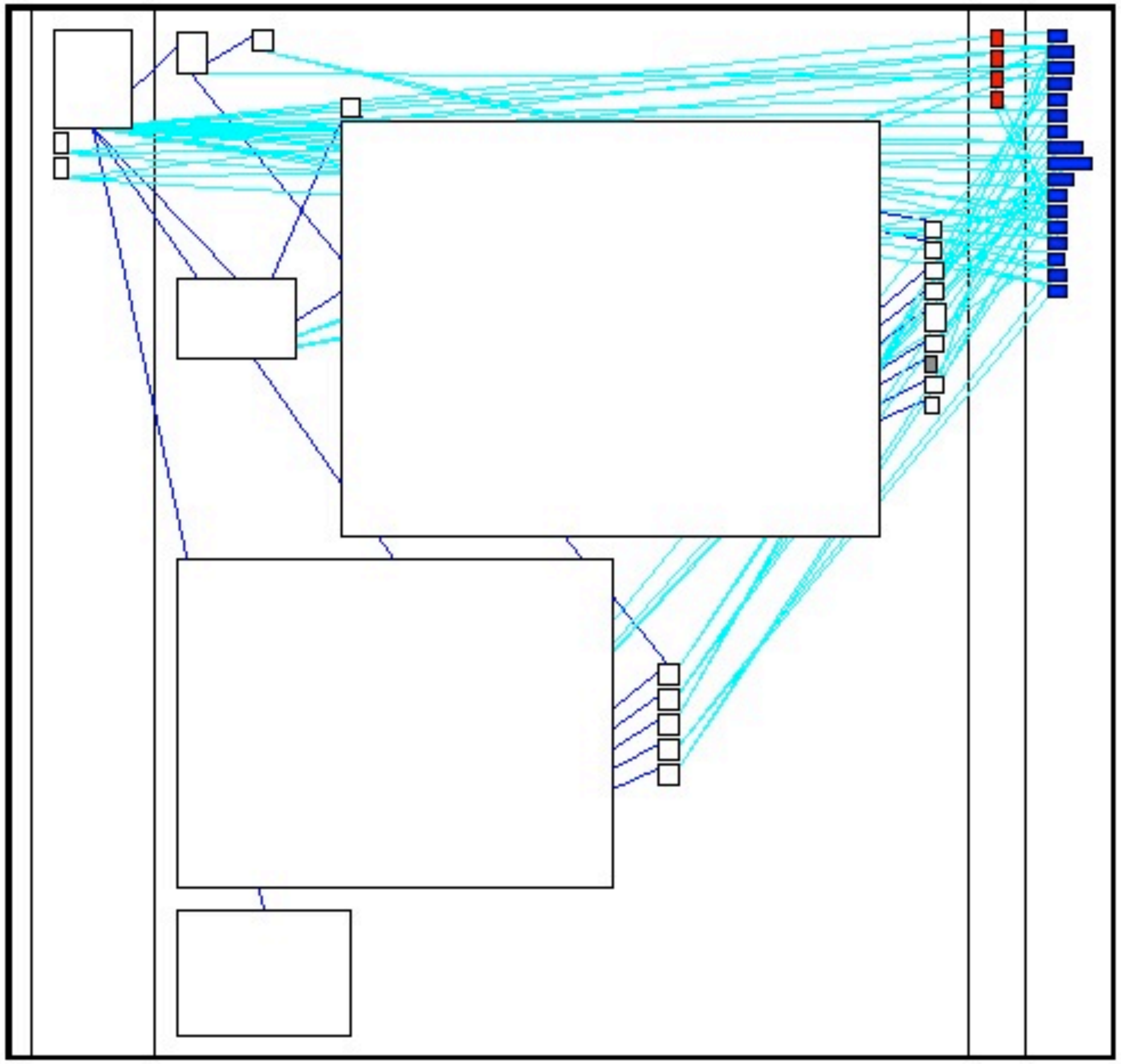


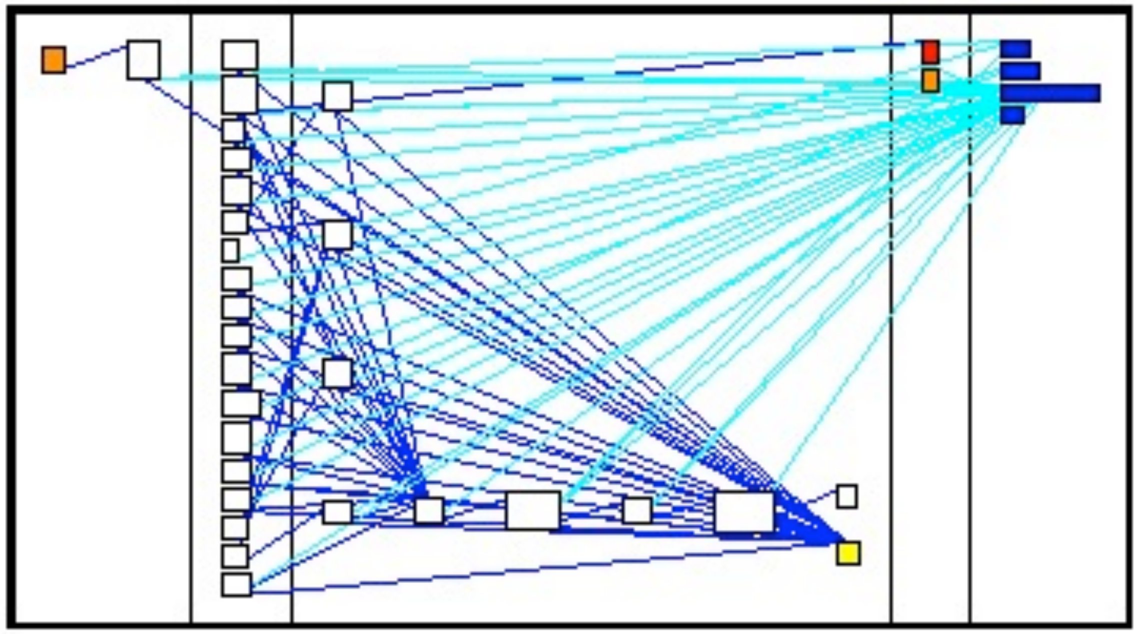
Class Blueprint shows class internals.







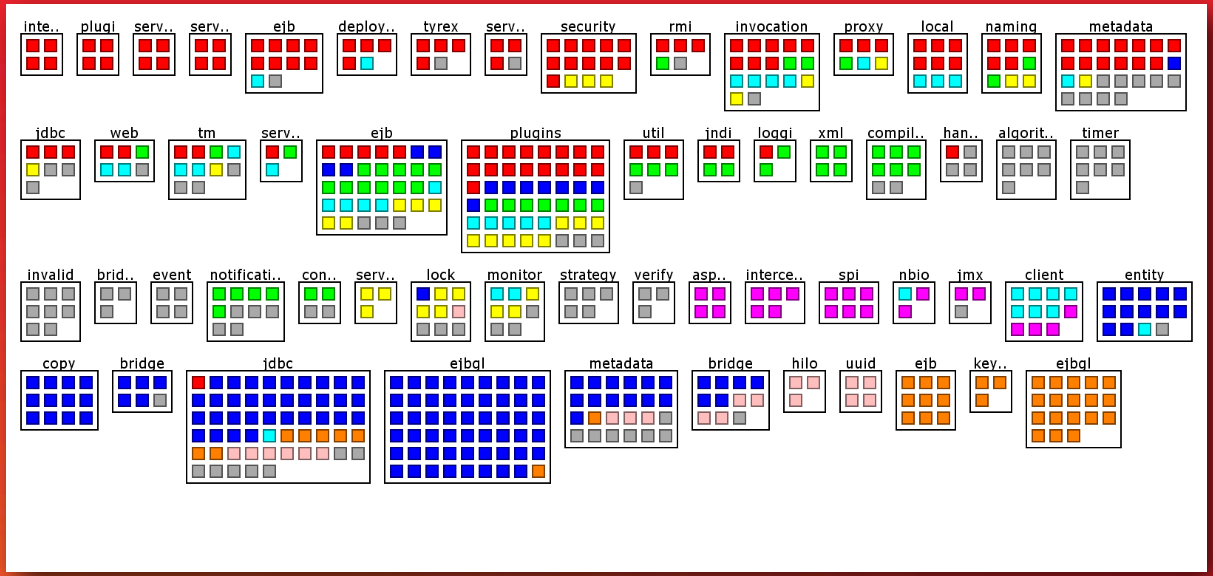




How properties spread on a system?

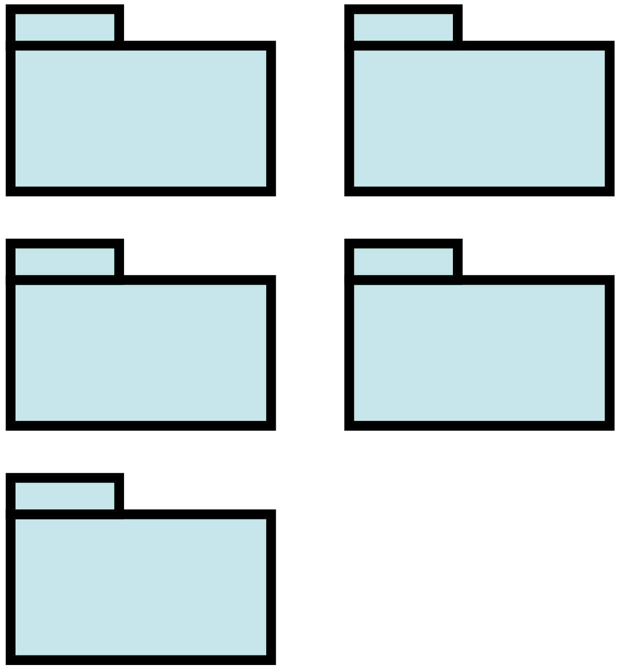
- Where author X worked?
- What are the classes under development the last two weeks?

- Distribution Map [ICSM]

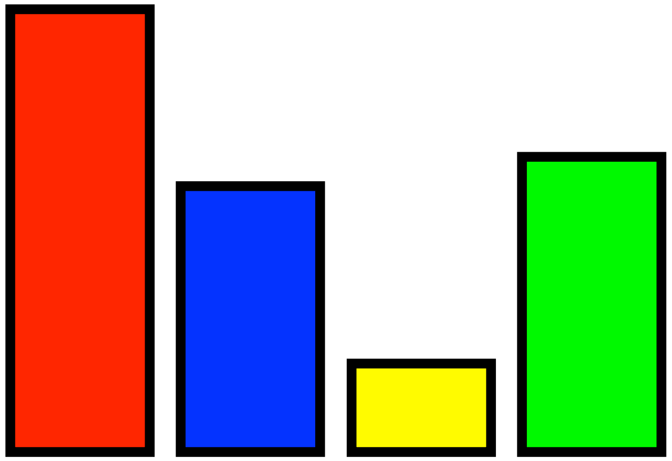


We take any two partitions, and

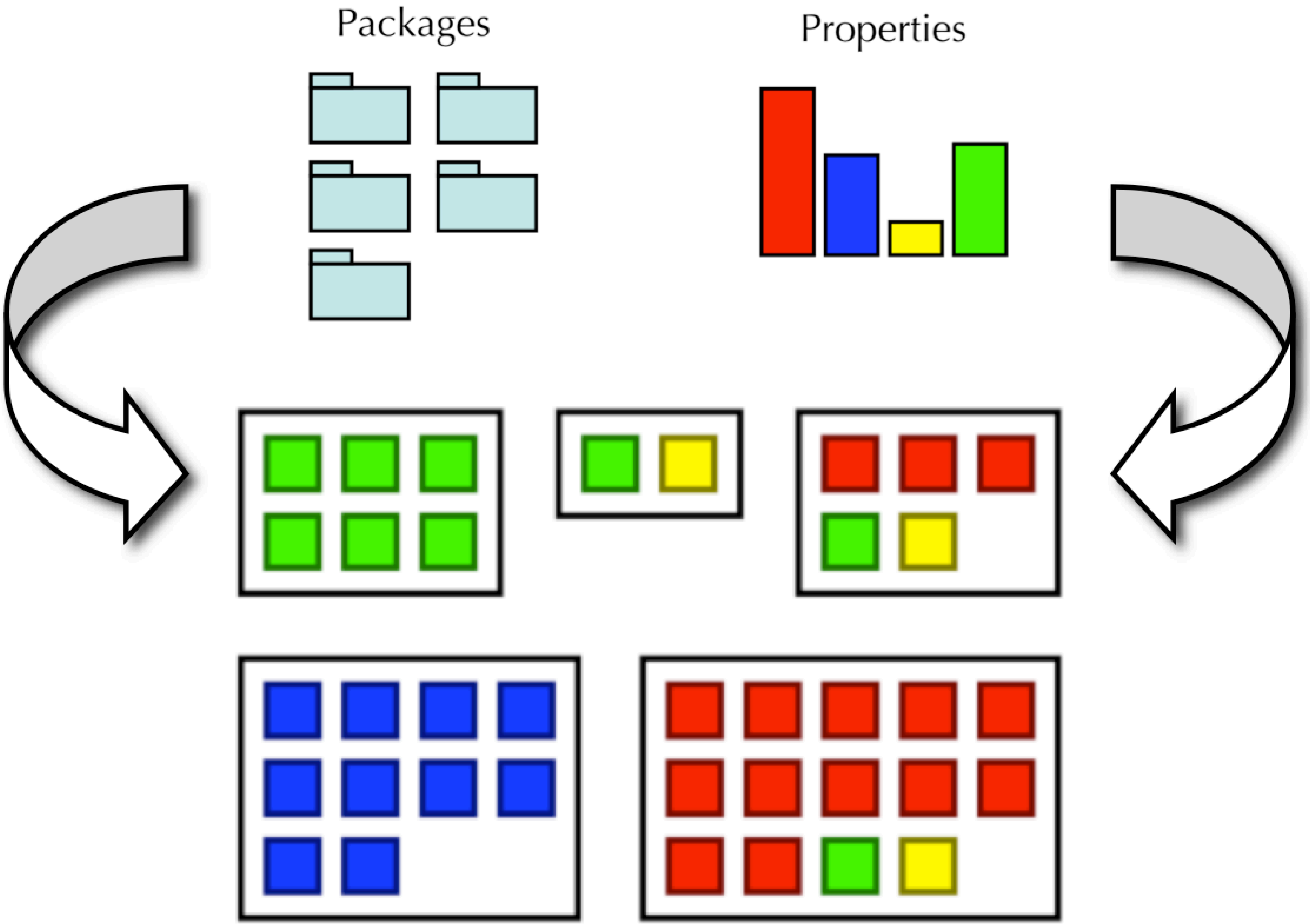
Packages



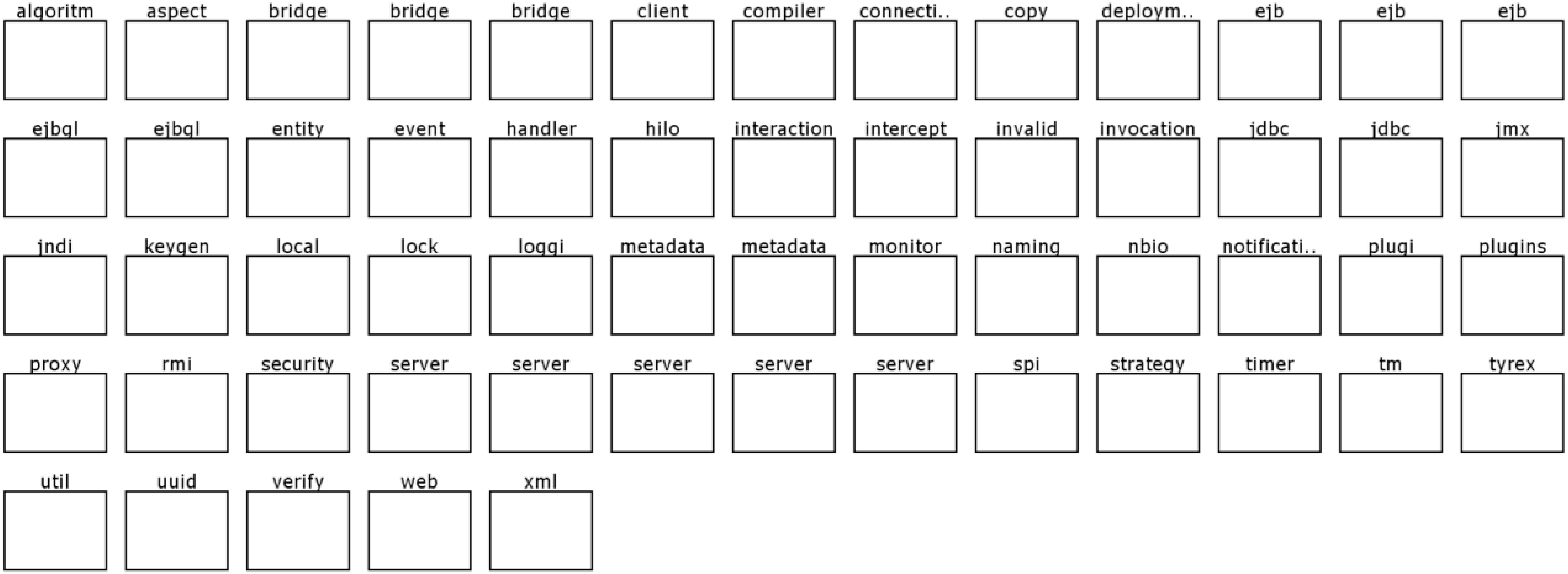
Properties



and create a Distribution Map.



Step 1 — for each package draw a rectangle



algoritm aspect bridge bridge bridge client compiler connecti.. copy deploym.. ejb ejb ejb

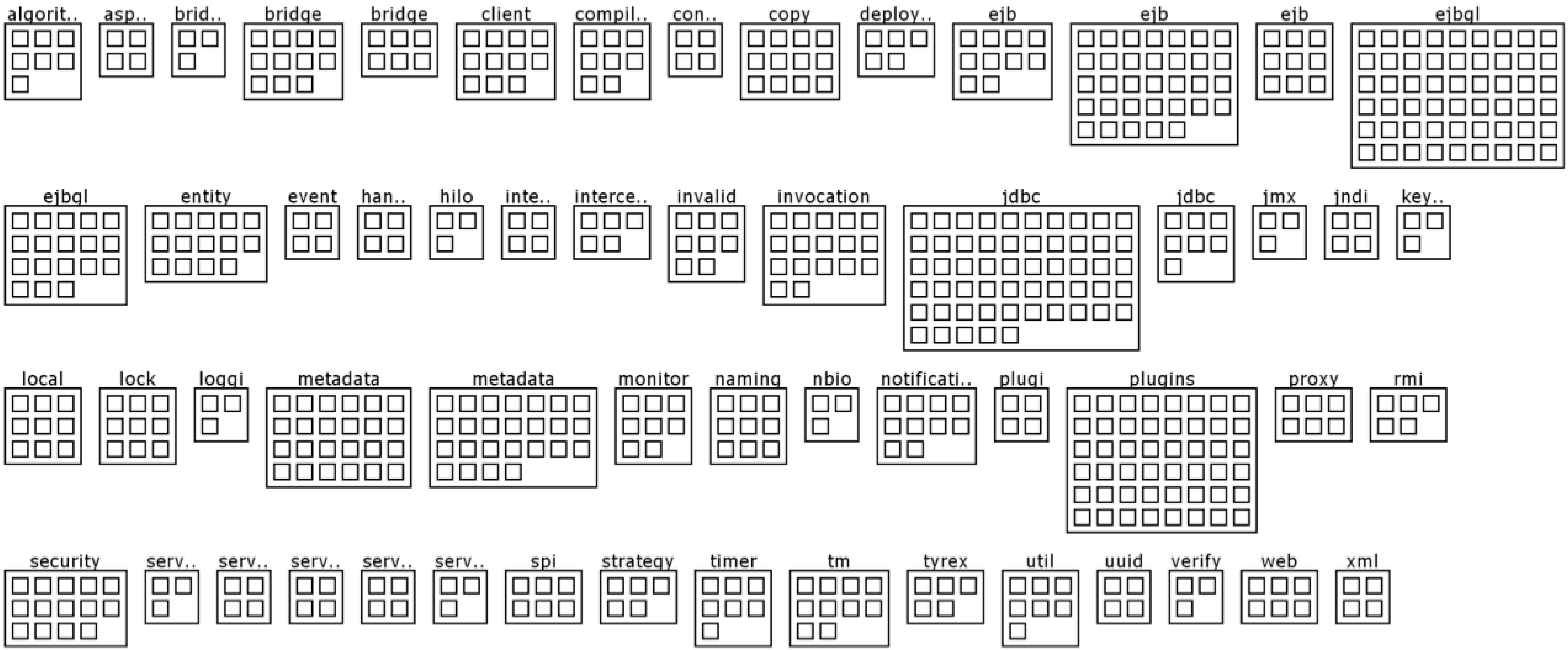
ejbql ejbql entity event handler hilo interaction intercept invalid invocation jdbc jdbc jmx

jndi keygen local lock loggi metadata metadata monitor naming nbio notificati.. plugi plugins

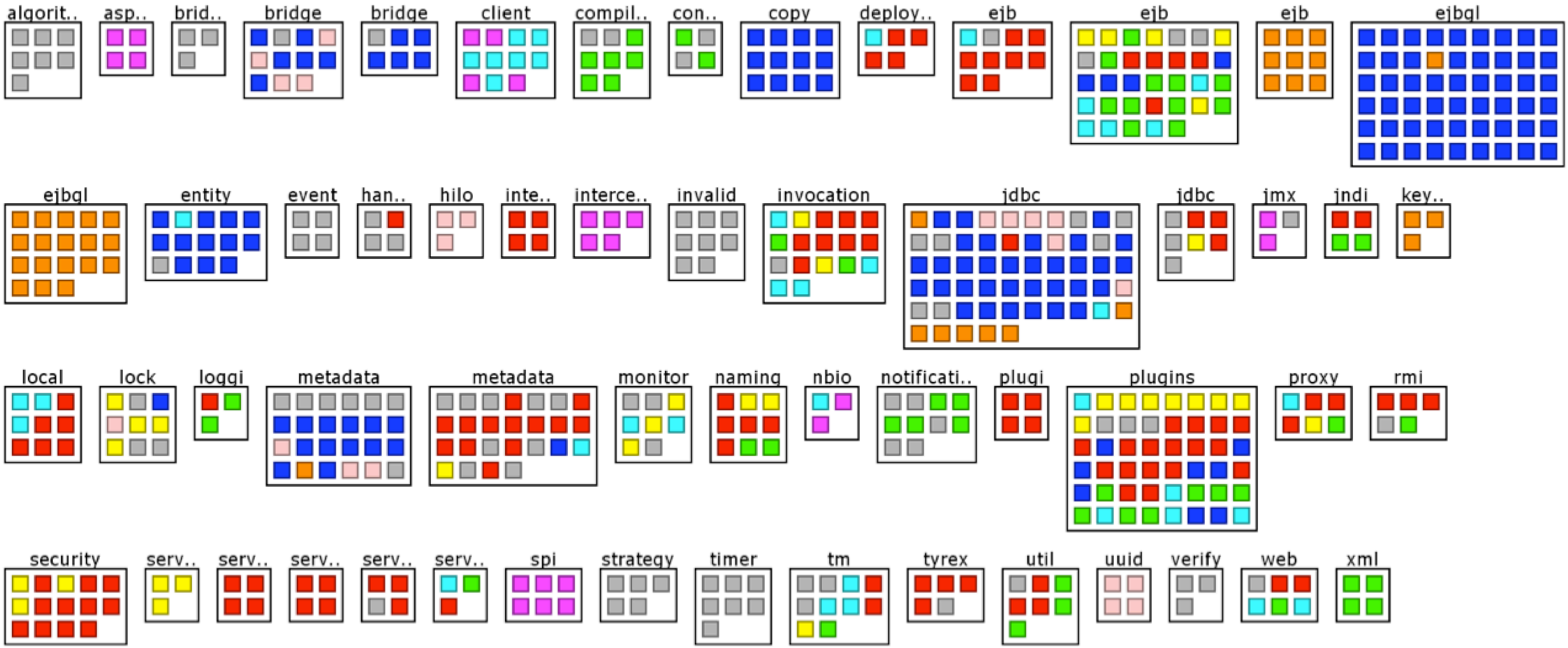
proxy rmi security server server server server server spi strategy timer tm tyrex

util uuid verify web xml

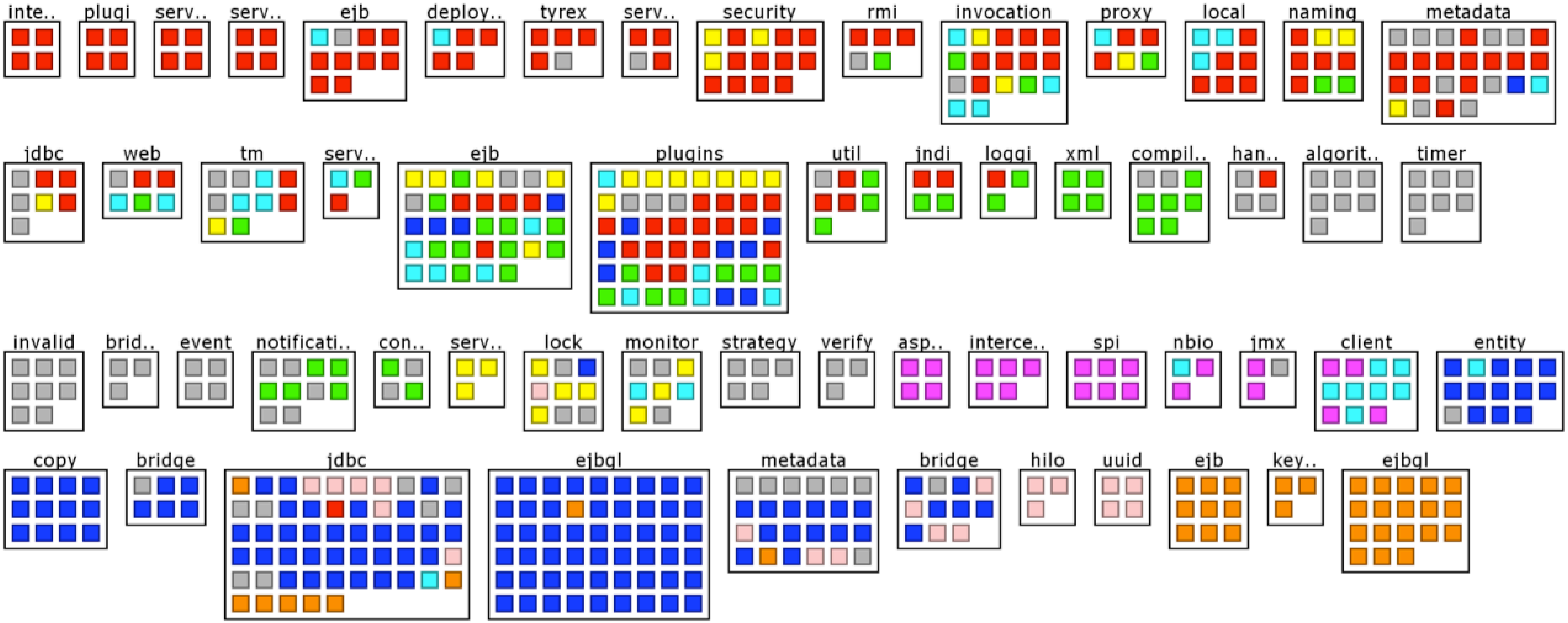
Step 2 — populate packages with classes



Step 3 — color the classes by property

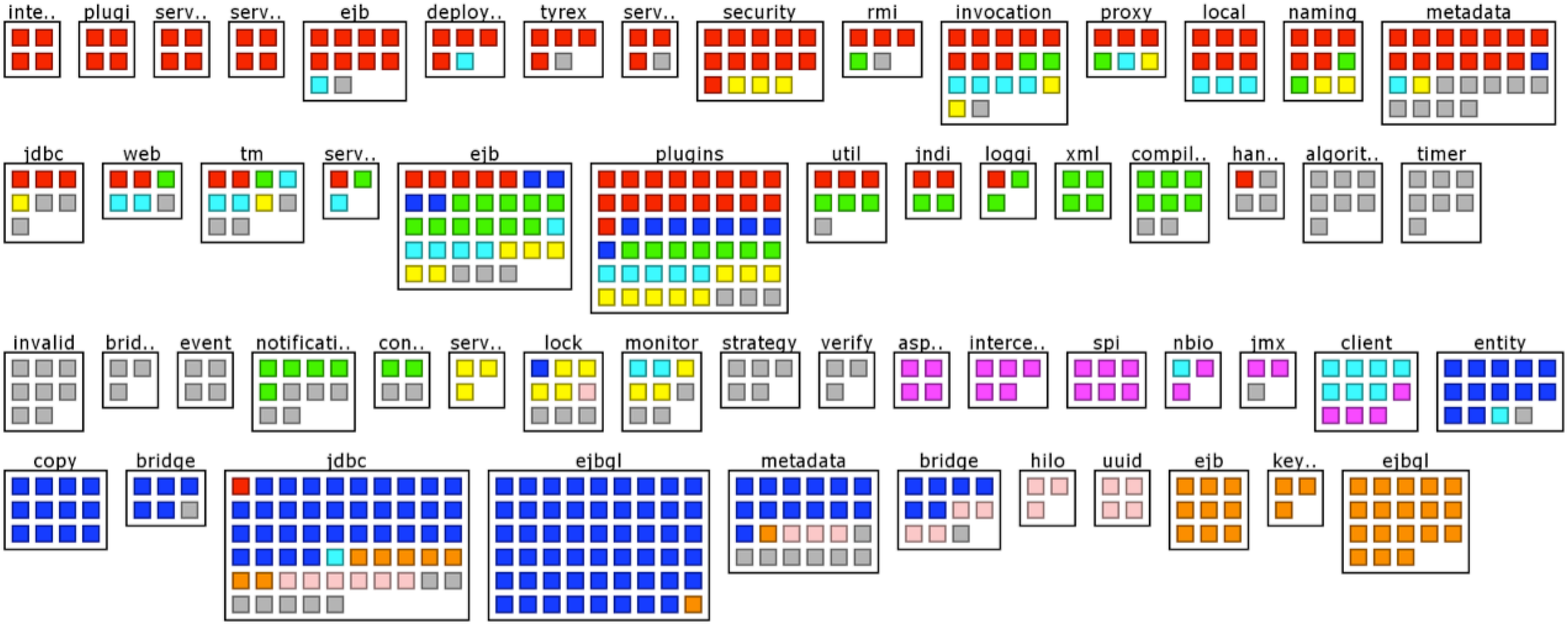


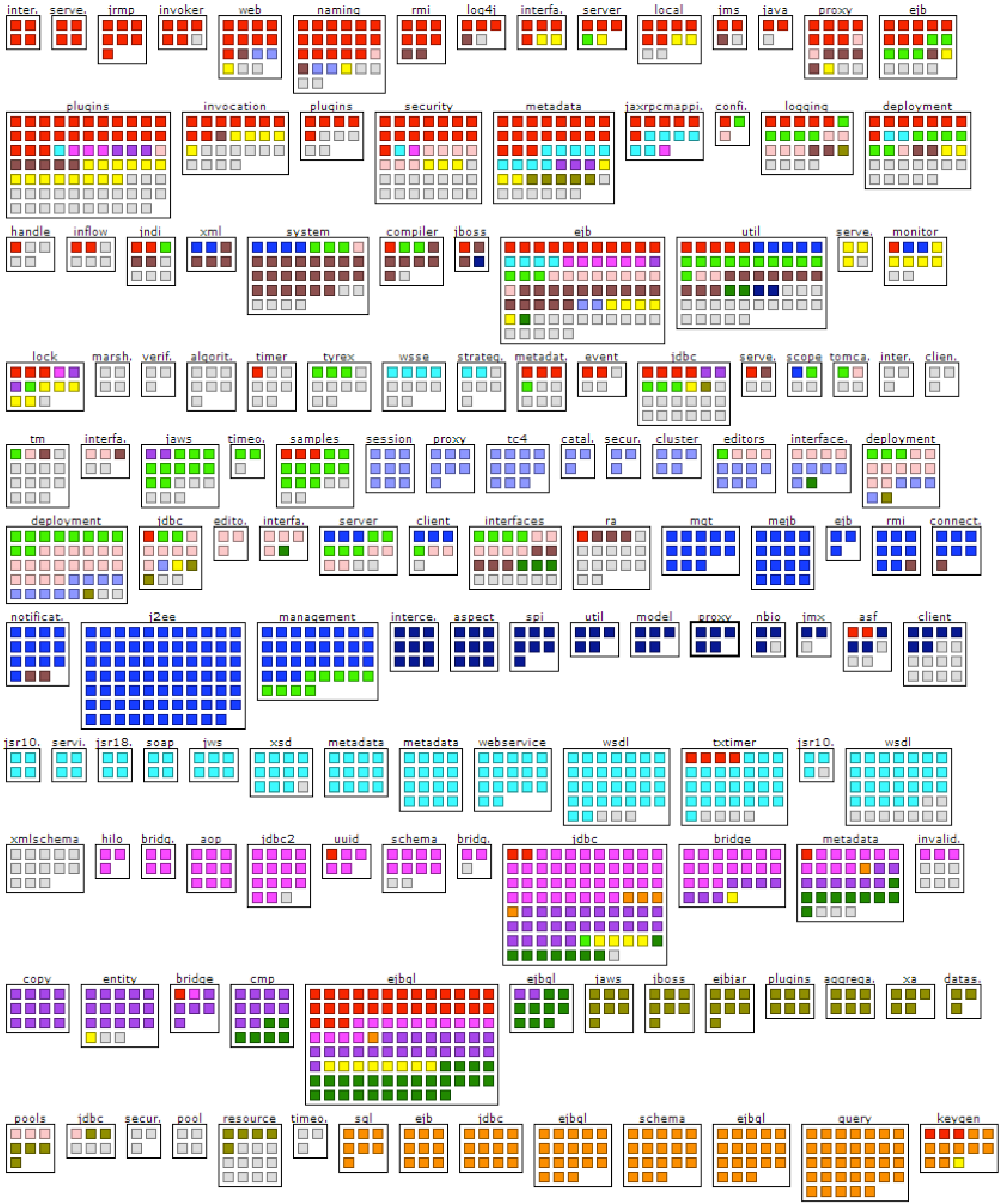
Step 4 — sort packages by content



Sorting with dendrogram seriation.

Step 5 — sort classes by properties

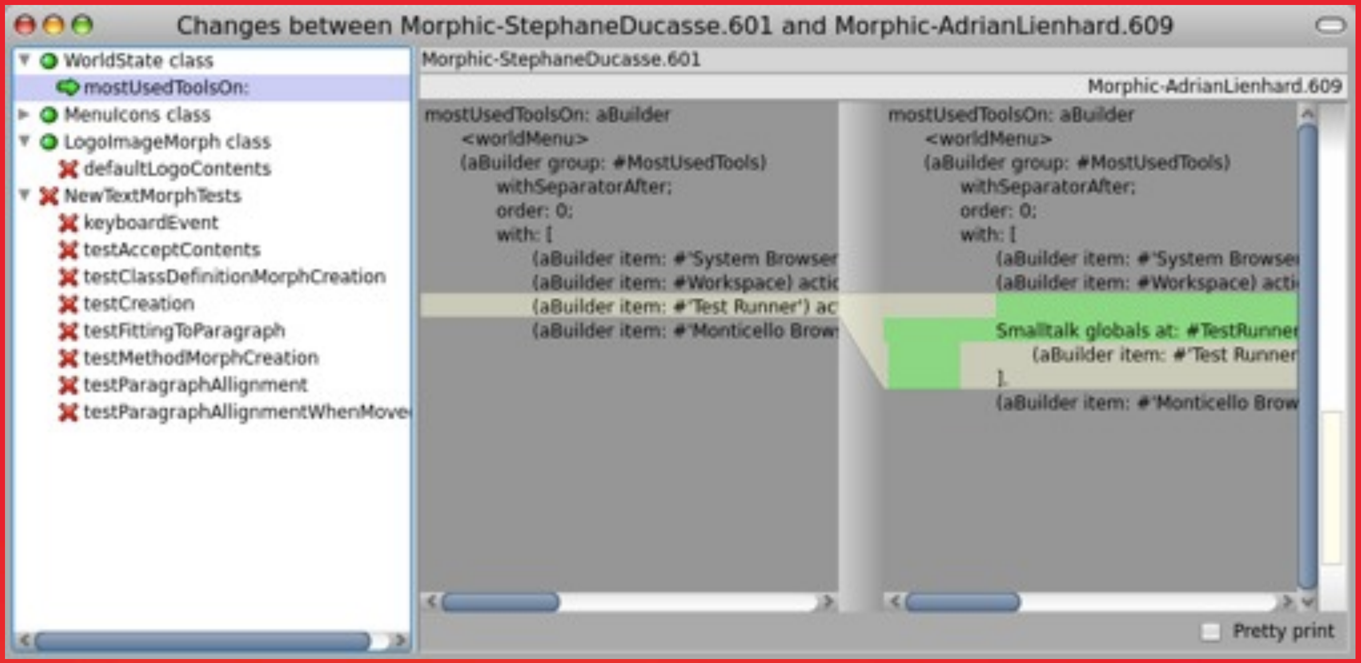



inter.
serve.
jrmp
invoker
web
naming
rmi
log4j
interfa.
server
local
jms
java
proxy
ejb
plugins
invocation
plugins
security
metadata
jaxrpcmappi.
confi.
logging
deployment
handle
inflow
jndi
xml
system
compiler
jboss
ejb
util
serve.
monitor
lock
marsh.
verif.
algorit.
timer
tyrex
wsse
strateg.
metadat.
event
jdbc
serve.
scope
tomca.
inter.
clien.
tm
interfa.
jaws
timeo.
samples
session
proxy
tc4
catal.
secur.
cluster
editors
interface.
deployment
deployment
jdbc
edito.
interfa.
server
client
interfaces
ra
mgt
mejb
ejb
rmi
connect.
notificat.
j2ee
management
interce.
aspect
spi
util
model
proxy
nbio
jmx
asf
client
jsr10.
servi.
jsr18.
soap
jws
xsd
metadata
metadata
webservice
wsdl
txtimer
jsr10.
wsdl
xmlschema
hilo
bridg.
aop
jdbc2
uuid
schema
bridg.
jdbc
bridge
metadata
invalid.
copy
entity
bridge
cmp
ejbql
ejbql
jaws
jboss
ejbjar
plugins
aggrega.
xa
datas.
pools
jdbc
secur.
pool
resource
timeo.
sql
ejb
jdbc
ejbql
schema
ejbql
query
keygen

How to understand changes

- Torch is the work of V. Uquillas-Gomez

How to understand changes

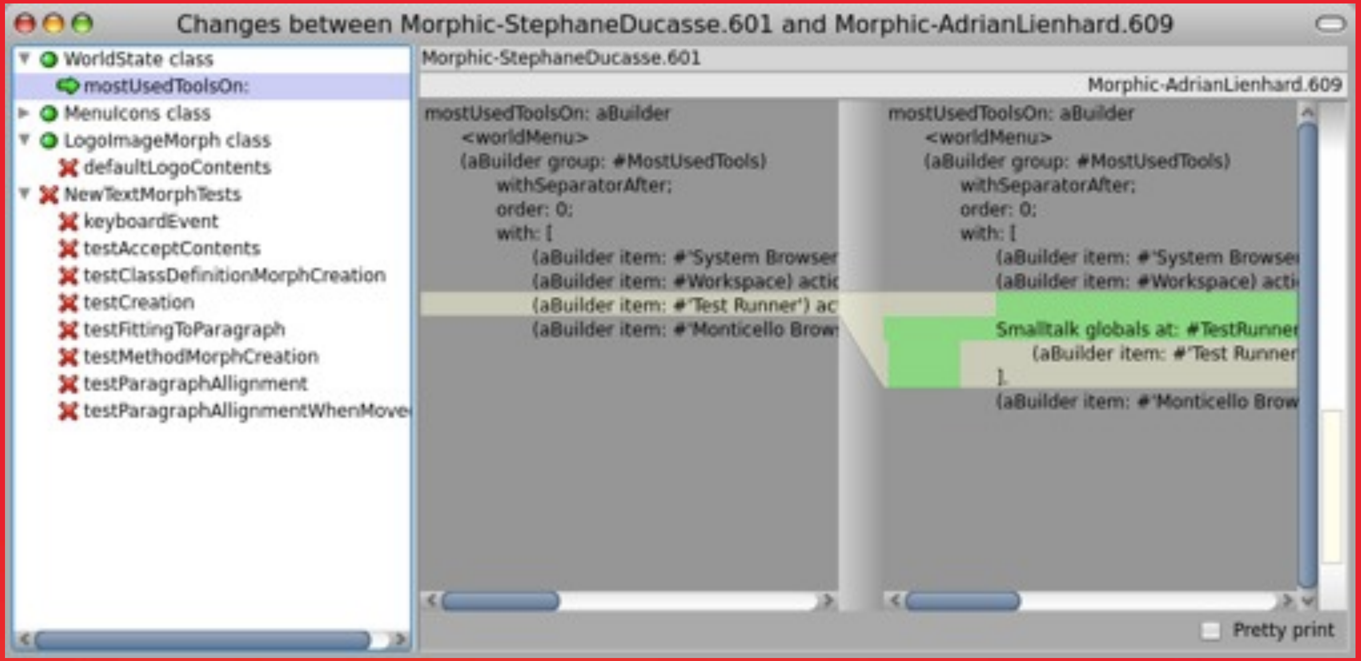


```
mostUsedToolsOn: aBuilder
    <worldMenu>
    (aBuilder group: #MostUsedTools)
        withSeparatorAfter;
        order: 0;
        with: [
            (aBuilder item: #'System Browser
            (aBuilder item: #Workspace) actic
            (aBuilder item: #'Test Runner') ac
            (aBuilder item: #'Monticello Brow
```

```
mostUsedToolsOn: aBuilder
    <worldMenu>
    (aBuilder group: #MostUsedTools)
        withSeparatorAfter;
        order: 0;
        with: [
            (aBuilder item: #'System Browse
            (aBuilder item: #Workspace) acti

            Smalltalk globals at: #TestRunner
                (aBuilder item: #'Test Runner
            ].
            (aBuilder item: #'Monticello Brow
```

- Torch is the work of V. Uquillas-Gomez

How to understand changes



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

```
▼ WorldState class
    mostUsedToolsOn:
▶ MenuIcons class
▼ LogoImageMorph class
    defaultLogoContents
▼ NewTextMorphTests
    keyboardEvent
    testAcceptContents
    testClassDefinitionMorphCreation
    testCreation
    testFittingToParagraph
    testMethodMorphCreation
    testParagraphAllignment
    testParagraphAllignmentWhenMove
```

Morphic-StephaneDucasse.601

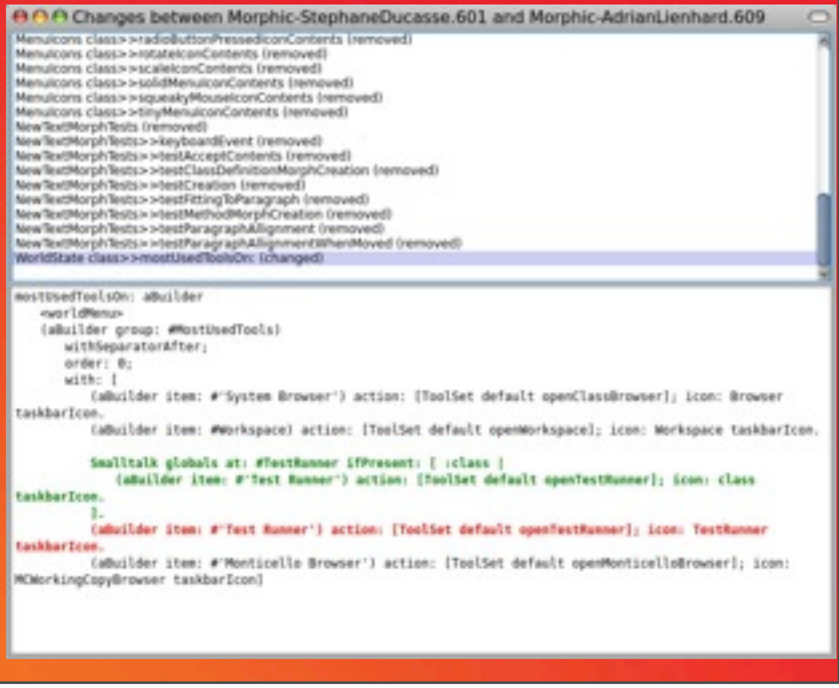
Morphic-AdrianLienhard.609

```
mostUsedToolsOn: aBuilder
    <worldMenu>
    (aBuilder group: #MostUsedTools)
        withSeparatorAfter;
        order: 0;
        with: [
            (aBuilder item: #'System Browser
            (aBuilder item: #Workspace) actic
            (aBuilder item: #'Test Runner') ac
            (aBuilder item: #'Monticello Brow
```

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

☐ Pretty print

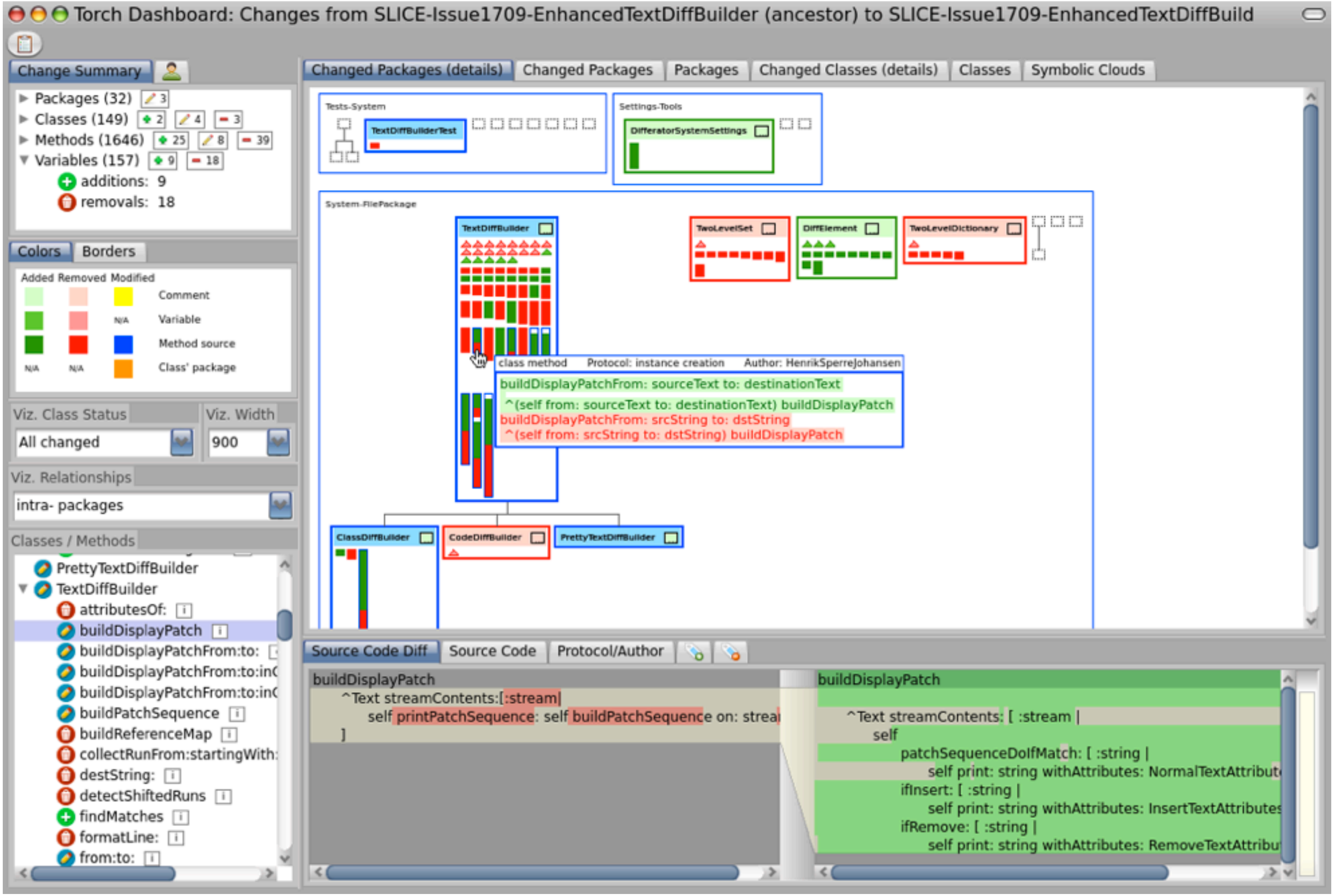
- Torch is the work of V. Uquillas-Gomez



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

```
MenuIcons class>>radioButtonPressedIconContents (removed)
MenuIcons class>>rotateIconContents (removed)
MenuIcons class>>scaleIconContents (removed)
MenuIcons class>>solidMenuIconContents (removed)
MenuIcons class>>squeakyMouseIconContents (removed)
MenuIcons class>>tinyMenuIconContents (removed)
NewTextMorphTests (removed)
NewTextMorphTests>>keyboardEvent (removed)
NewTextMorphTests>>testAcceptContents (removed)
NewTextMorphTests>>testClassDefinitionMorphCreation (removed)
NewTextMorphTests>>testCreation (removed)
NewTextMorphTests>>testFittingToParagraph (removed)
NewTextMorphTests>>testMethodMorphCreation (removed)
NewTextMorphTests>>testParagraphAllignment (removed)
NewTextMorphTests>>testParagraphAllignmentWhenMoved (removed)
WorldState class>>mostUsedToolsOn: (changed)
```

```
mostUsedToolsOn: aBuilder
    <worldMenu>
    (aBuilder group: #MostUsedTools)
        withSeparatorAfter;
        order: 0;
        with: [
            (aBuilder item: #'System Browser') action: [ToolSet default openClassBrowser]; icon: Browser
taskbarIcon.
            (aBuilder item: #Workspace) action: [ToolSet default openWorkspace]; icon: Workspace taskbarIcon.

            Smalltalk globals at: #TestRunner ifPresent: [ :class |
                (aBuilder item: #'Test Runner') action: [ToolSet default openTestRunner]; icon: class
taskbarIcon.
            ].
            (aBuilder item: #'Test Runner') action: [ToolSet default openTestRunner]; icon: TestRunner
taskbarIcon.
            (aBuilder item: #'Monticello Browser') action: [ToolSet default openMonticelloBrowser]; icon:
MCWorkingCopyBrowser taskbarIcon]
```



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

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

Added	Removed	Modified	
🟩	🟥	🟨	Comment
🟩	🟥	N/A	Variable
🟩	🟥	🟦	Method source
N/A	N/A	🟧	Class' package

Viz. Class Status: All changed

Viz. Width: 900

Viz. Relationships: intra- packages

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

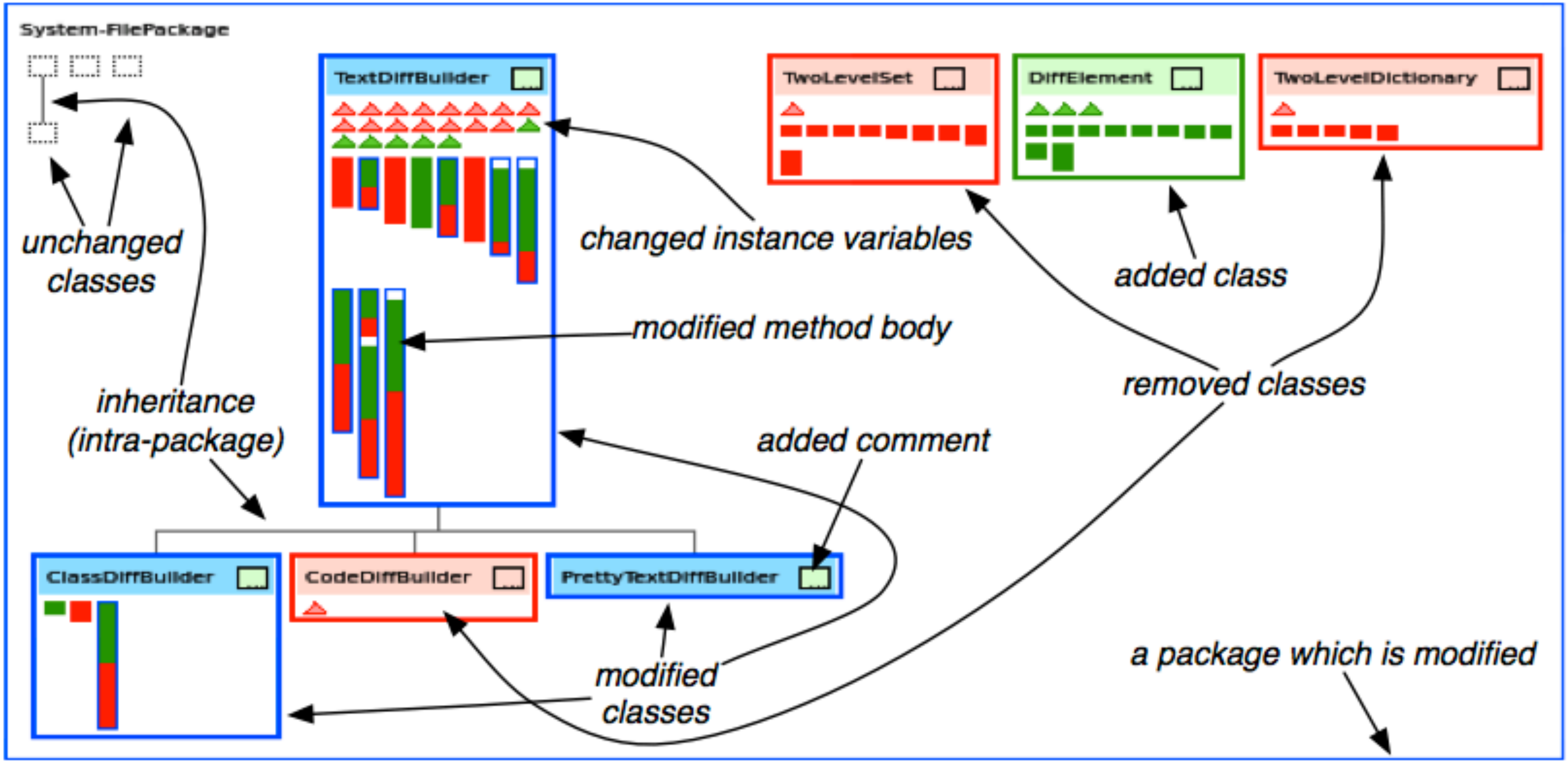
class method Protocol: instance creation Author: HenrikSperreJohansen

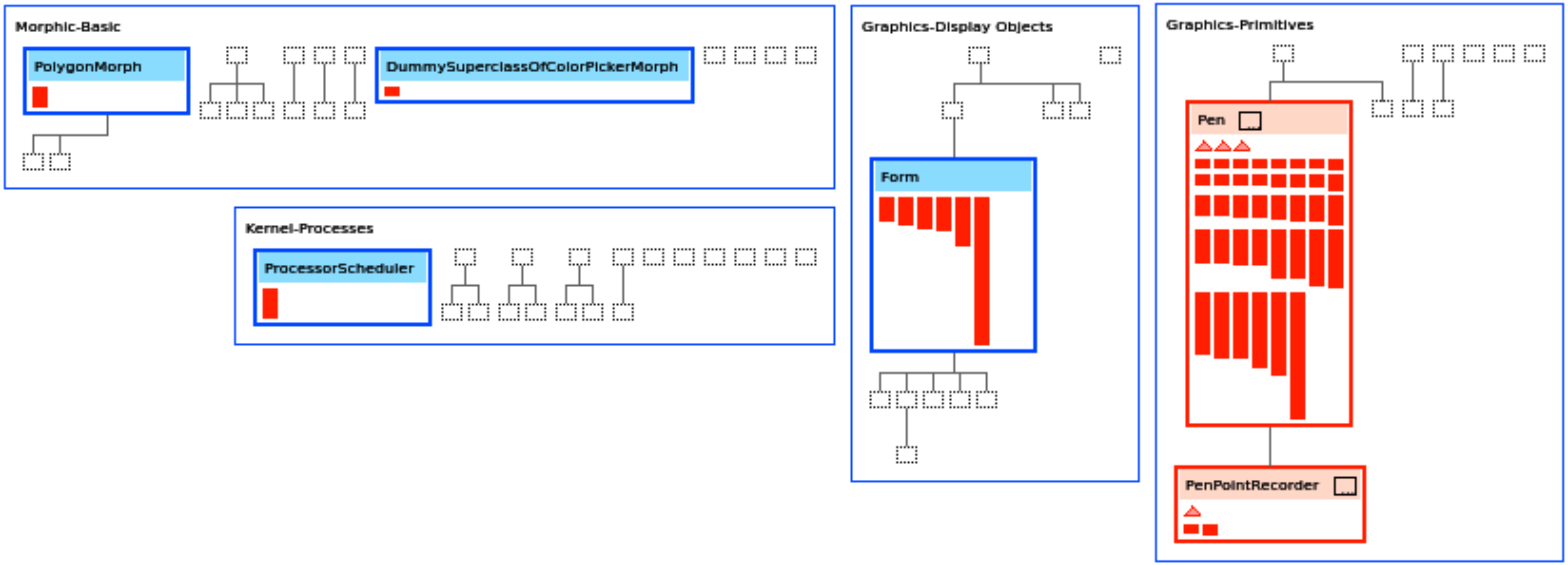
```
buildDisplayPatchFrom: sourceText to: destinationText

 ^(self from: sourceText to: destinationText) buildDisplayPatch
buildDisplayPatchFrom: srcString to: dstString
 ^(self from: srcString to: dstString) buildDisplayPatch
```

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

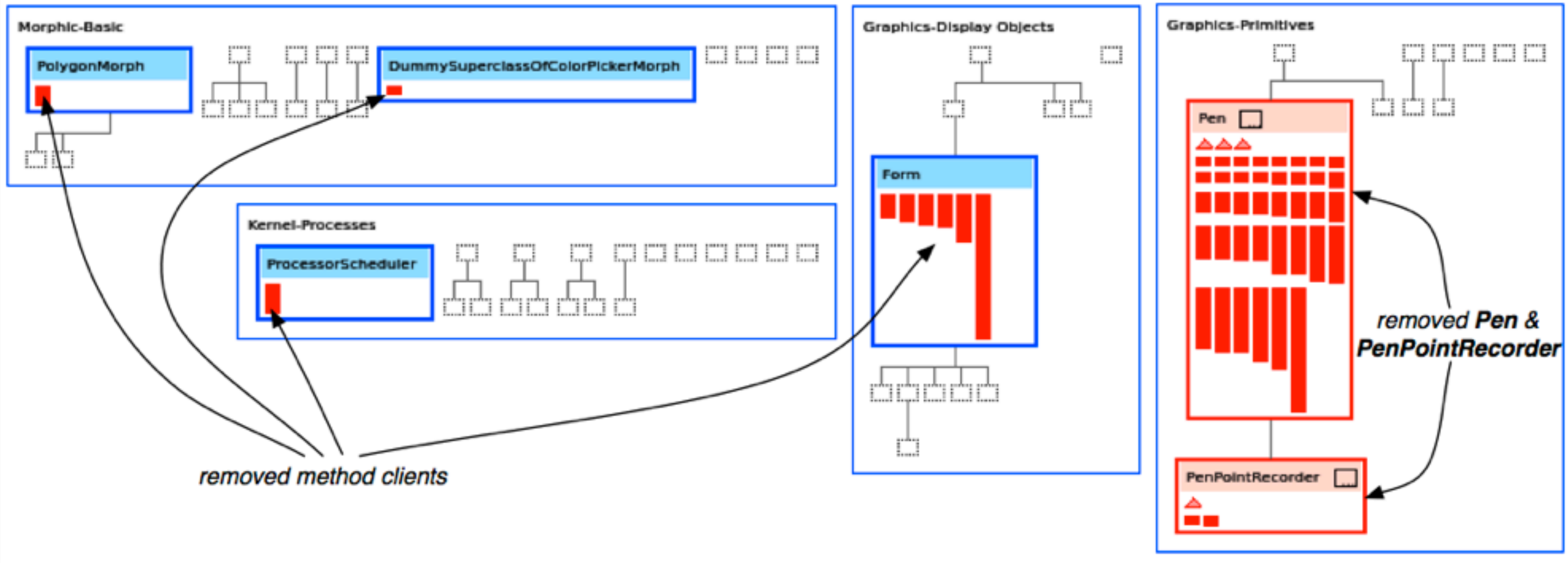
```
buildDisplayPatch

    ^Text streamContents: [ :stream |
        self
            patchSequenceDoIfMatch: [ :string |
                self print: string withAttributes: NormalTextAttribut
            ifInsert: [ :string |
                self print: string withAttributes: InsertTextAttributes
            ifRemove: [ :string |
                self print: string withAttributes: RemoveTextAttribu
```

Package Structure



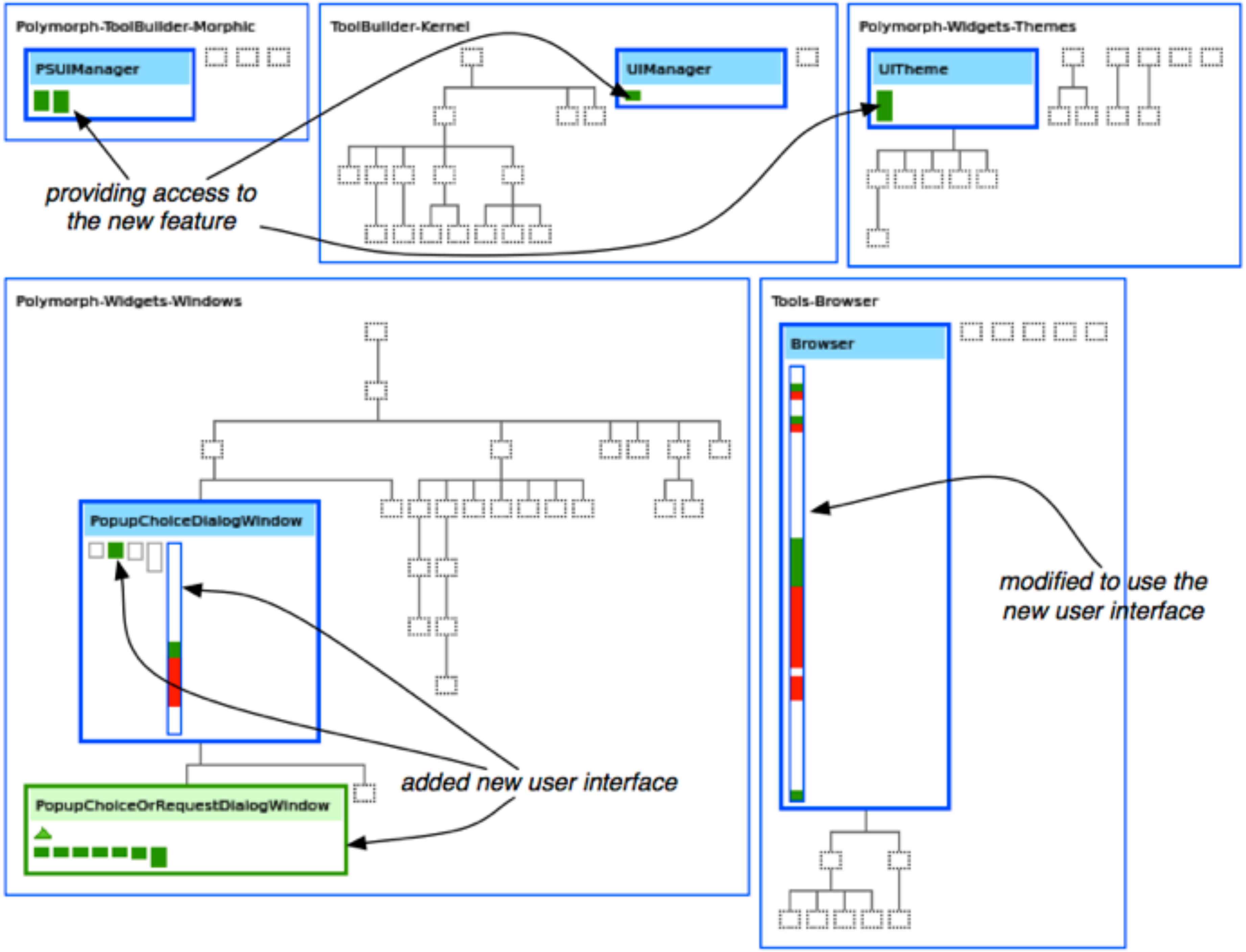

Morphic-Basic
PolygonMorph
DummySuperclassOfColorPickerMorph
Kernel-Processes
ProcessorScheduler
Graphics-Display Objects
Form
Graphics-Primitives
Pen
PenPointRecorder

Removing a feature (I)



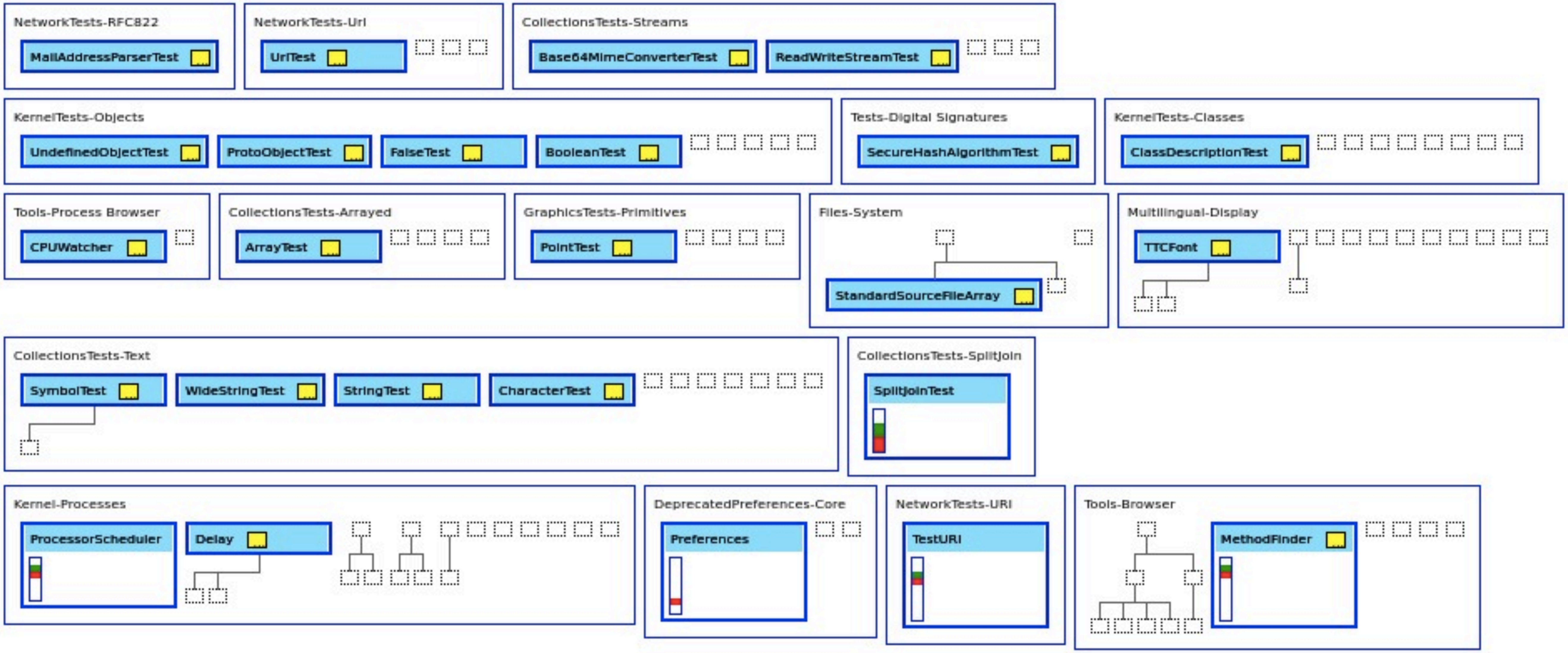
Removed **Pen** and **PenPointRecorder**

Introducing a feature


Polymorph-ToolBuilder-Morphic
PSUIManager
ToolBuilder-Kernel
UIManager
Polymorph-Widgets-Themes
UITheme
providing access to the new feature
Polymorph-Widgets-Windows
PopupChoiceDialogWindow
PopupChoiceOrRequestDialogWindow
added new user interface
Tools-Browser
Browser
modified to use the new user interface

Introduced **PopupChoice** / **RequestDialog**

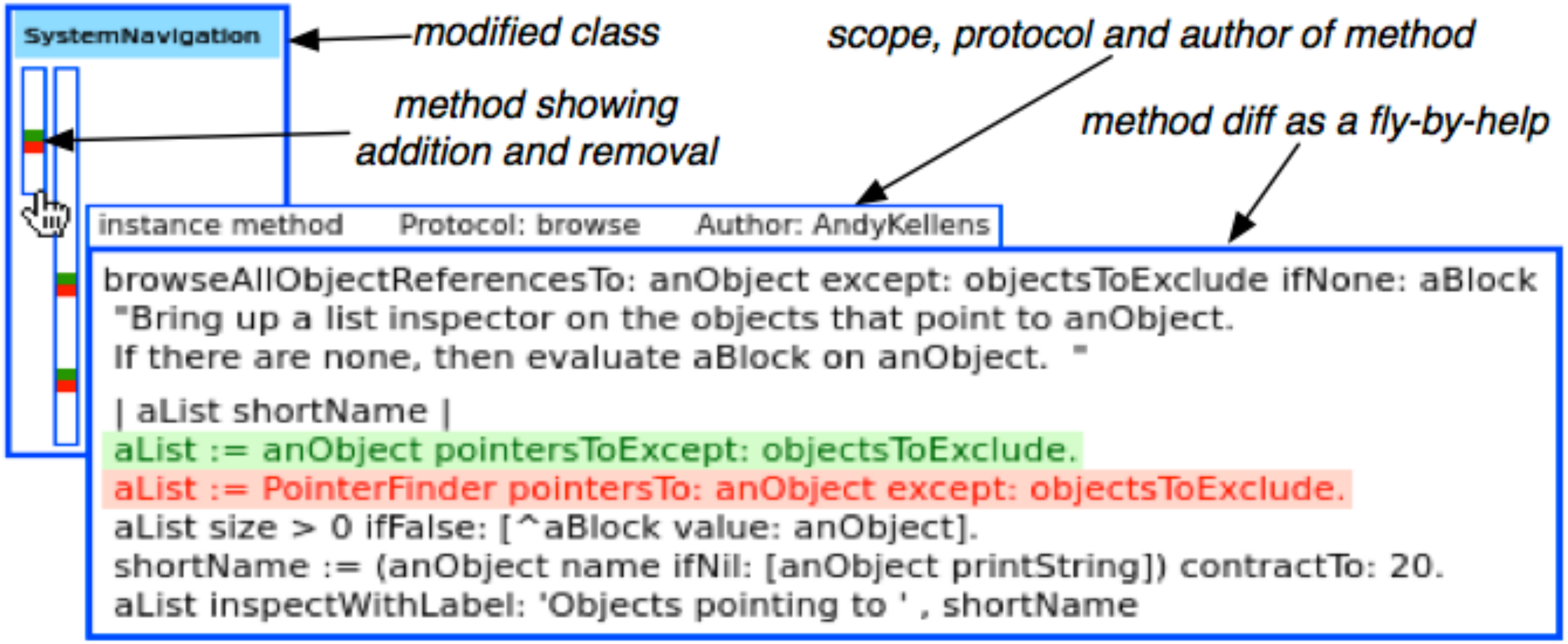
Editing comments



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


SystemNavigation
modified class
scope, protocol and author of method
method showing addition and removal
method diff as a fly-by-help

instance method Protocol: browse Author: AndyKellens

```
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
```

RMod

http://stephane.ducasse.free.fr