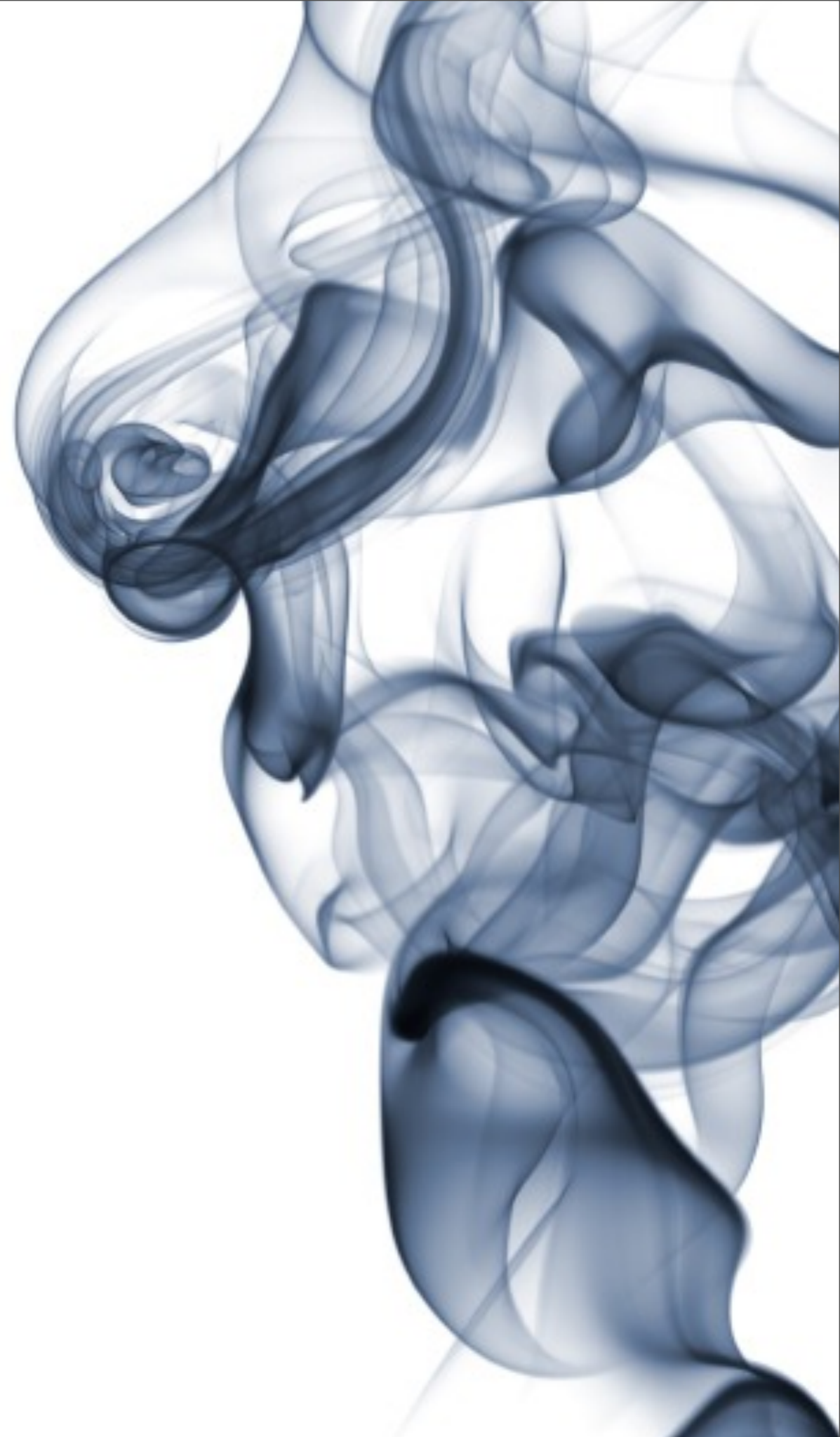


Software Evolution Analysis & Visualization

Ball's Dilemma

*Software is **intangible**,
having no physical
shape or size.*

Thomas Ball



How much Software is out there?

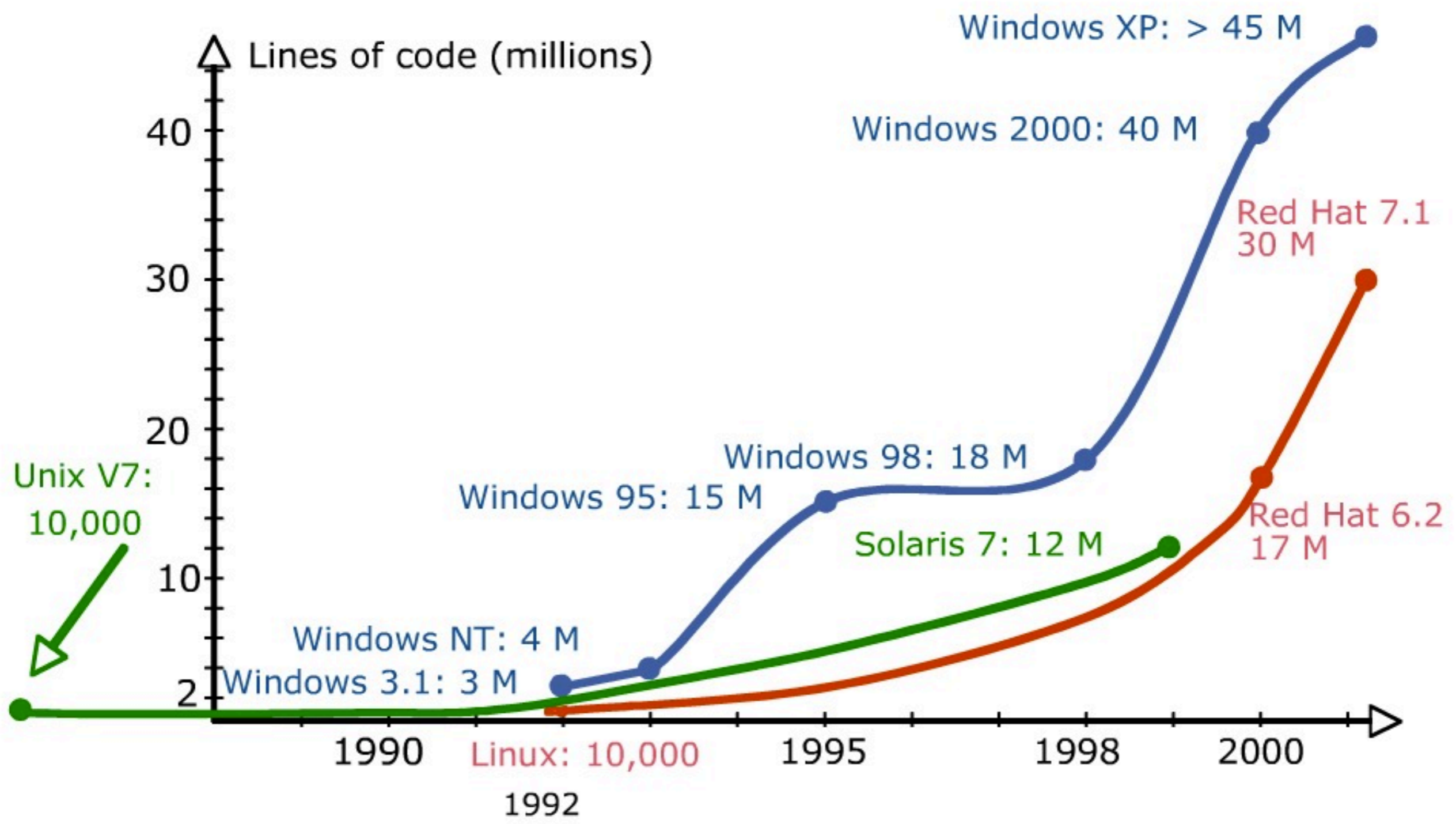
The total volume of software is estimated at 7'000'000'000 function points (FP)

- 1 FP ~ 128 lines of C or 107 lines of COBOL
- This means ca. 1 TLOC (1'000'000'000'000 lines)
- Printed on paper, we can wrap the planet 10 times!

In what shape is it?

- On average ca. 5 bugs / FP
- 35'000'000'000 bugs (6 per Person)





The growth of operating systems over time

Welcome to Microsoft...

...and your first task is to get familiar with our code base!

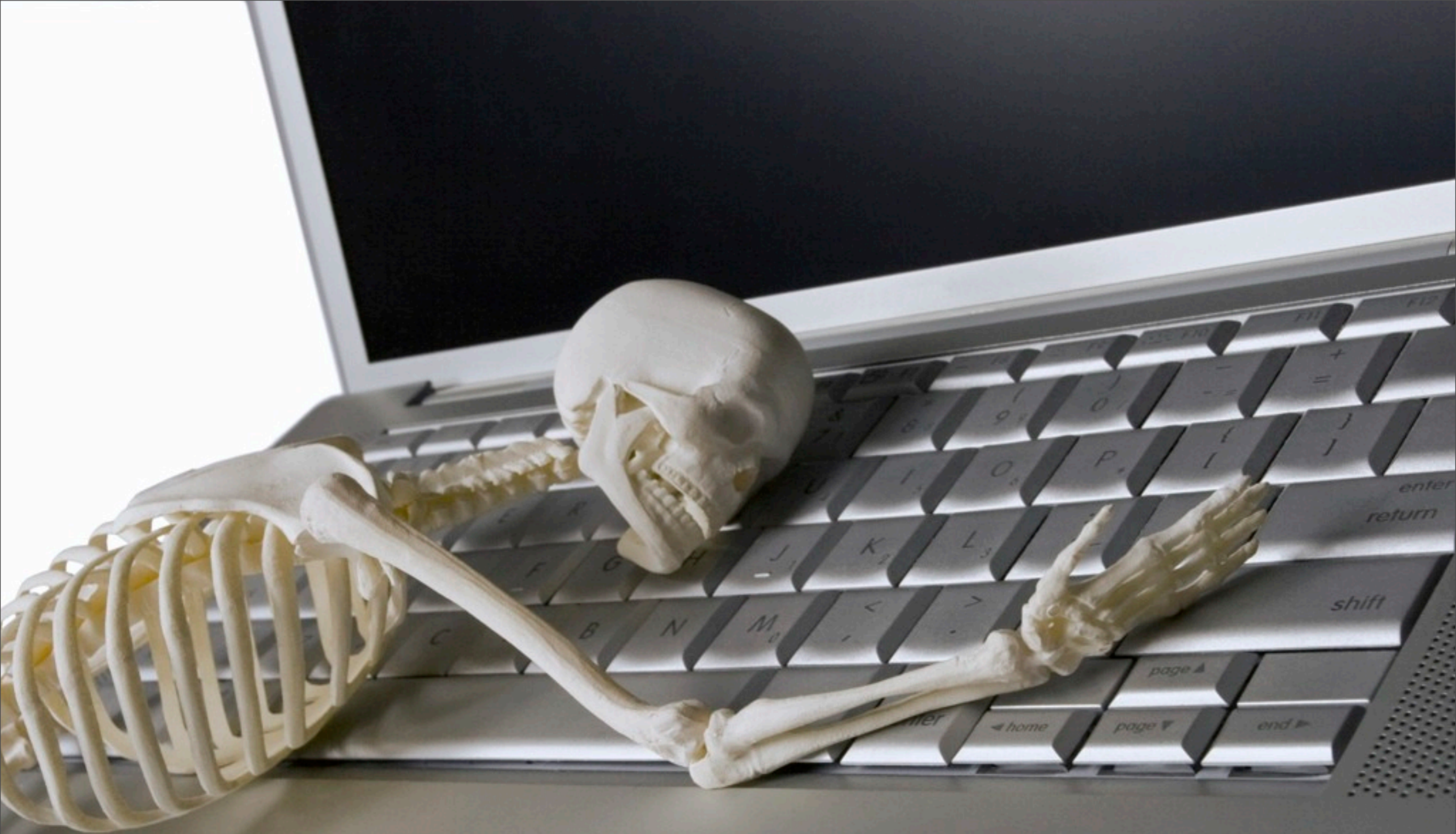
- Reverse engineer ~50 MLOC of Windows Vista Code
- 2 sec/line avg. reading speed = 100'000'000 sec
- / 3600 = 27'777 hours
- / 8 = 3472 days
- / 5 = 694 weeks

Roughly **14 to 15 years** just to read all the code!





Yes ~~we~~ ^{you} can!



By the way, you should also browse the change logs of the last few years to understand where we are coming from...

Software Evolution

*Software Evolution is the process of continual fixing, adaptation, enhancement to **maintain stakeholder satisfaction**.*

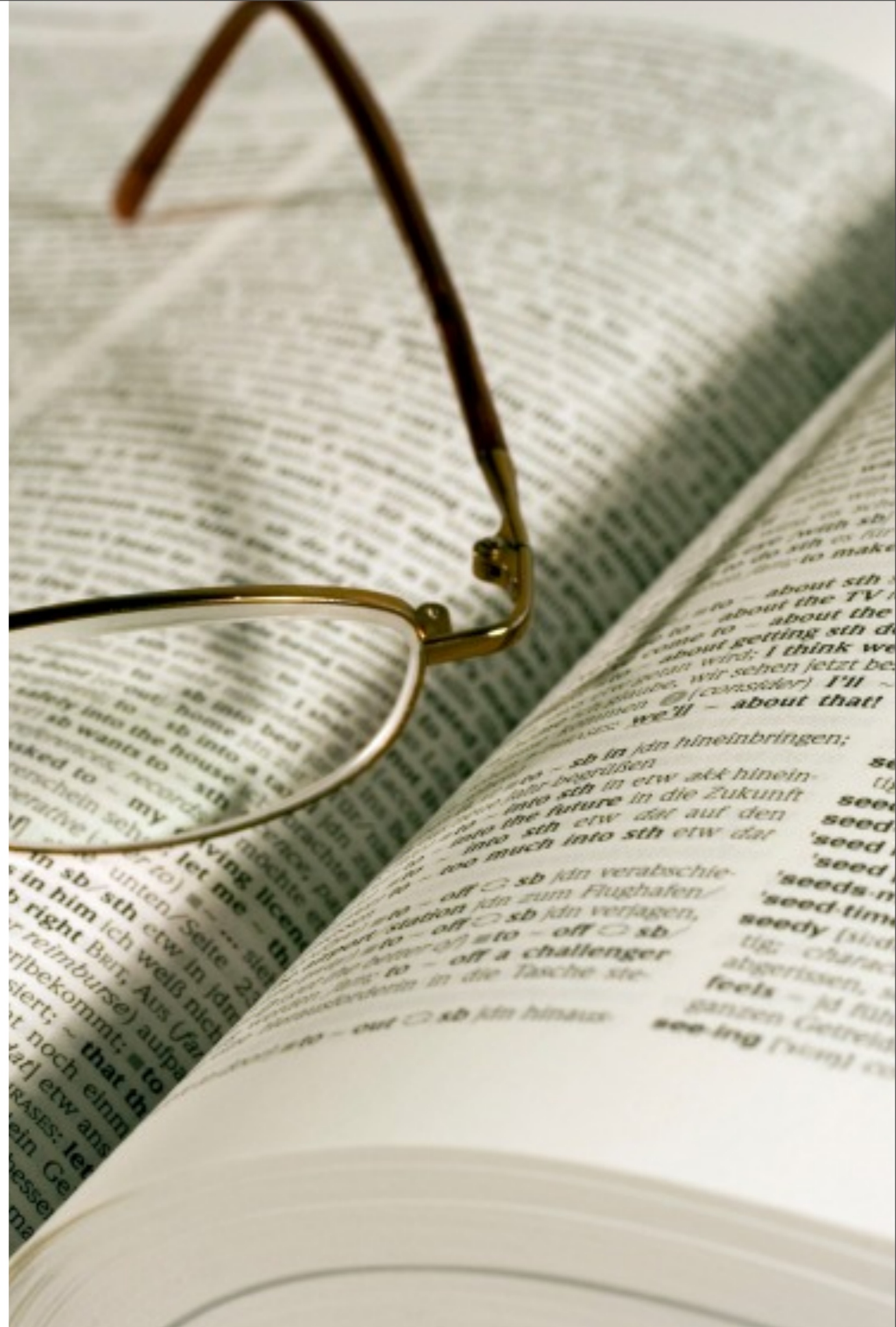
Lehman and Ramil

***Maintenance** means general **post-delivery** activities, whereas **Evolution** refers to a particular phase in the staged model where **substantial changes** are made to the software.*

Bennet and Rajlich

***Evolution is what happens** while you are busy making other plans.*

Godfrey



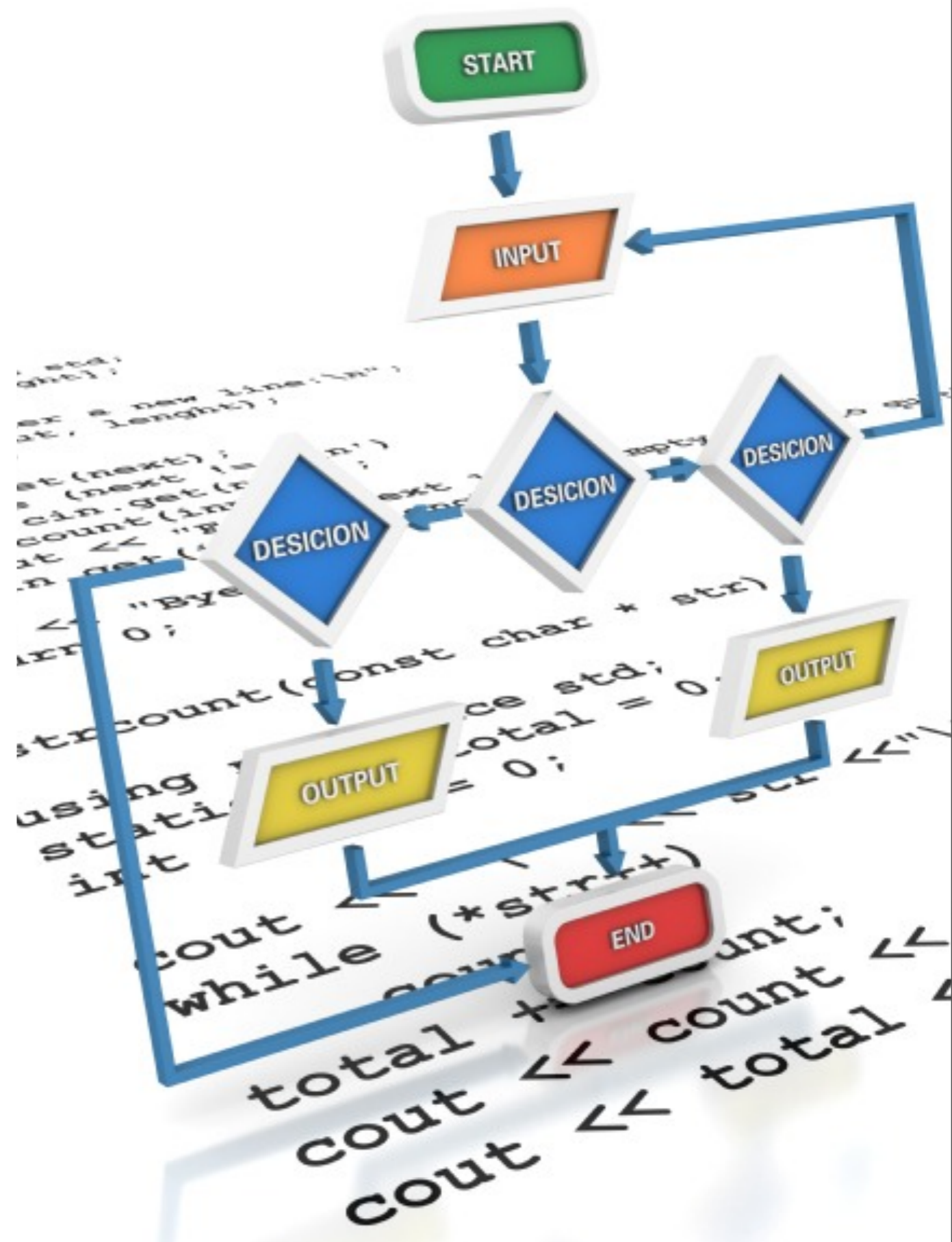
3 Types of Programs

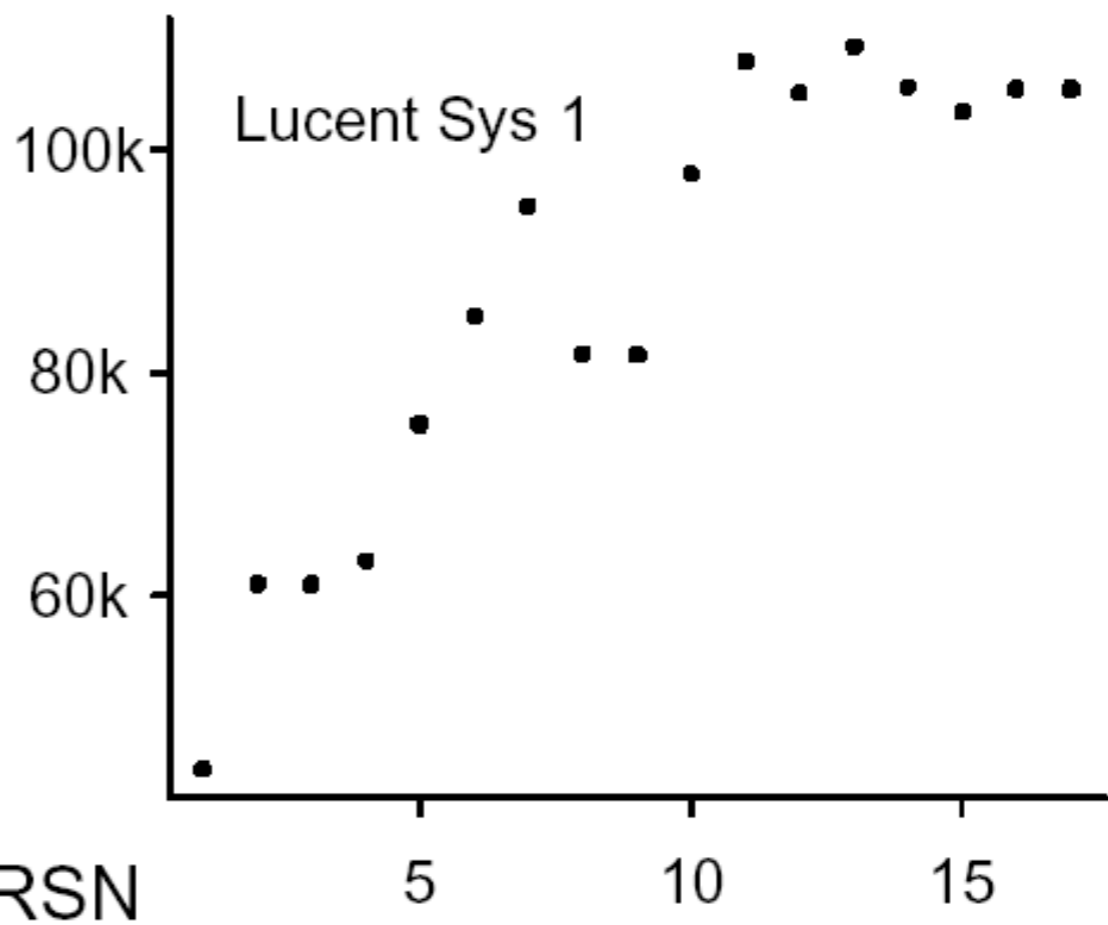
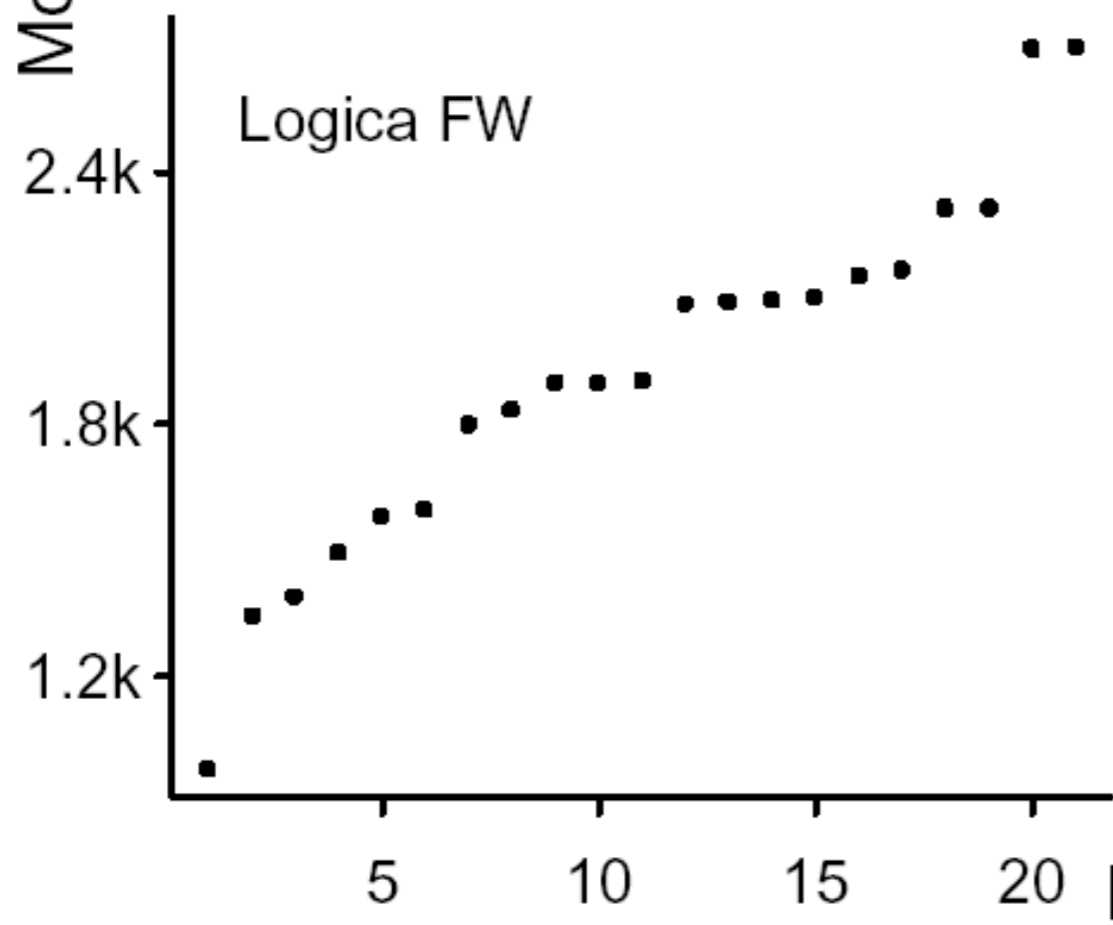
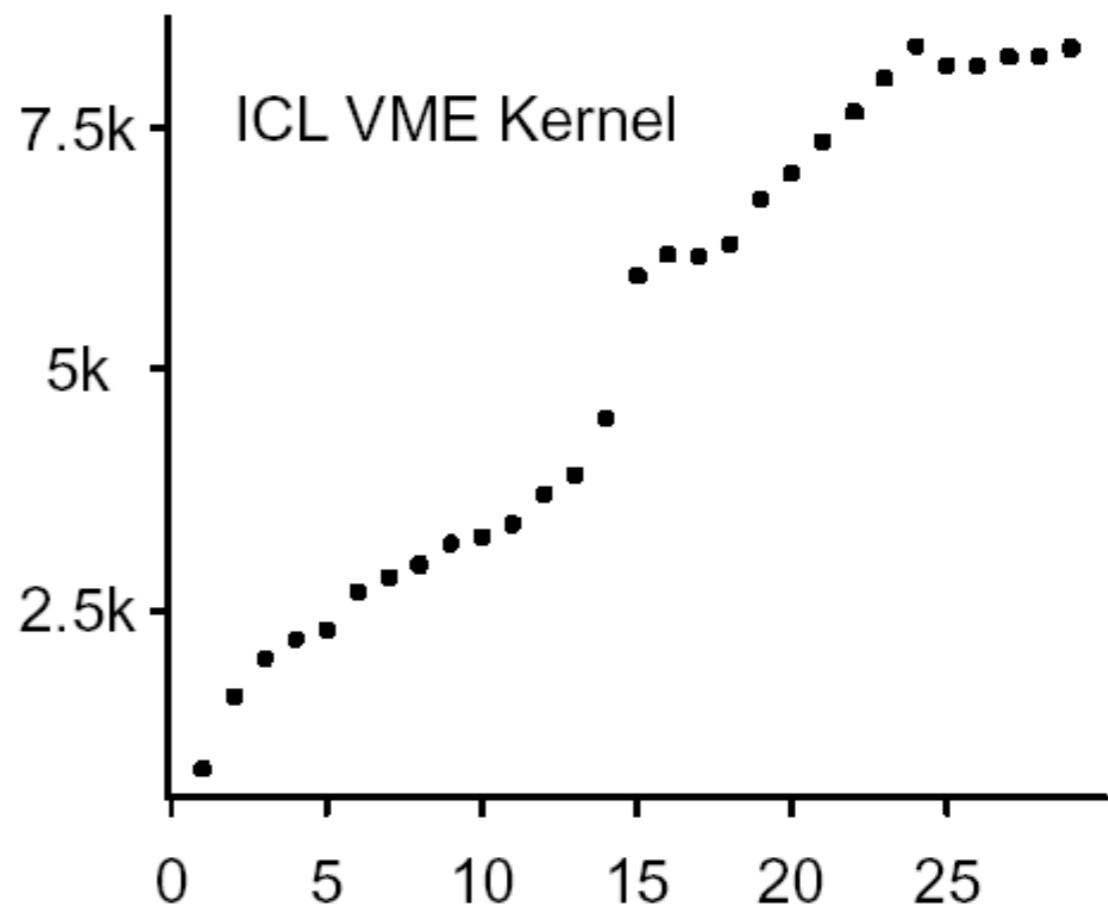
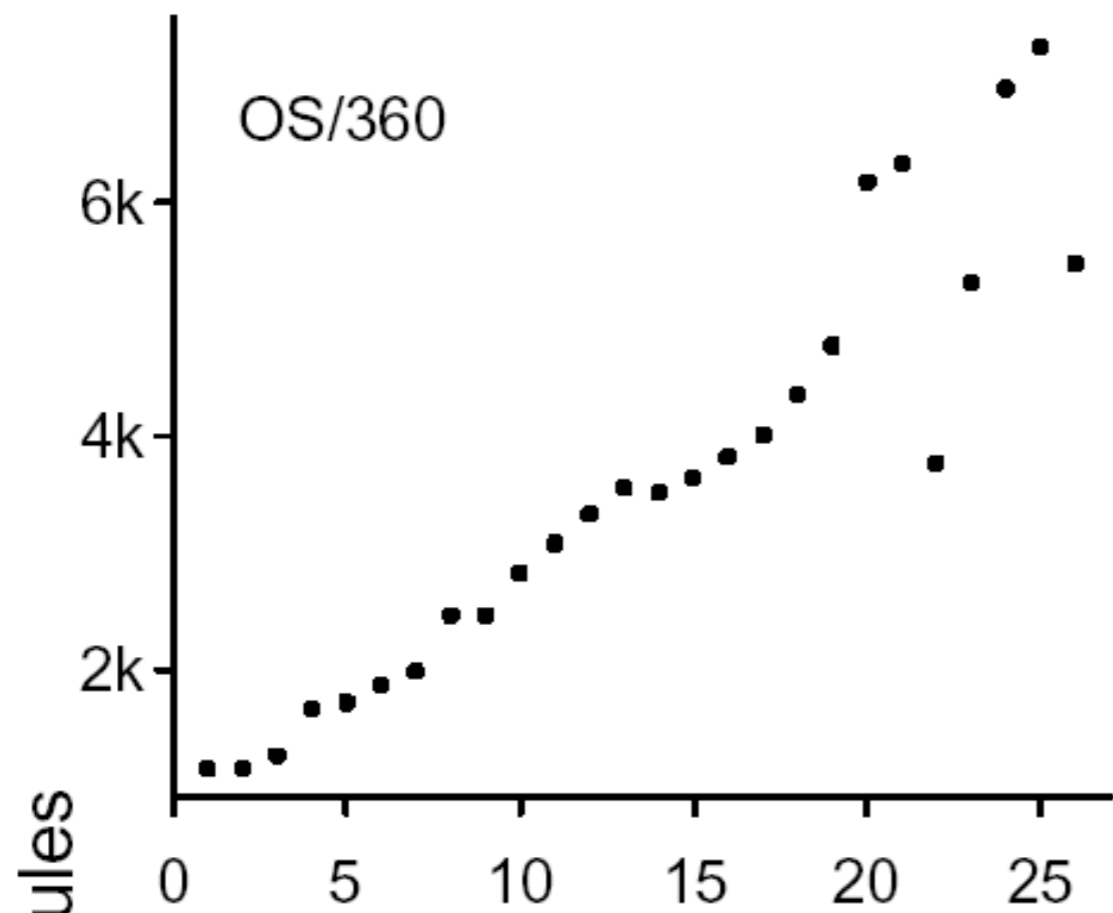
S-Type (“**Specifiable**”)
do not evolve

P-Type (“**Problem Solving**”)
are likely to evolve

E-Type (“**Embedded**”)
are inherently evolutionary, due to feedback loops

(Lehman and Belady, 1980)





Release History Database

<http://evolizer.org>

Integrates various software repositories in one query-able database:

- Source Code Management
- Issue Tracking
- Developer Mailinglists

Augments this data with:

- Links
- Fine grained changes
- Metrics




[Home](#)
[Mining Challenge](#)
[Data Track](#)
[Program](#)
[Venue](#)
[History](#)
[Organization](#)


News

Mar 19, 2013

Notifications are out, and the list of accepted papers is now available.

Mar 14, 2013

Brian Doll is a keynote speaker at MSR 2013!

Mar 4, 2013

The extension has now passed. Thank you, everyone, for your mining challenge submissions. **MSR Challenge 24hr Extension**—Submissions for the MSR Challenge will be accepted until 11:59PM Pacific time on March 5th, 2013, for a 24-hour extension.

Feb 13, 2013

The extension has now passed. Thank you, everyone, for your submissions. **Deadline Extension!!!**—We will allow authors to update their **Research, Practice and Data** submissions up to (and including) February 18th, 2013. An initial submission must be made by the original deadline.

Jan 14, 2013

This year's prize for the best Mining Challenge will be a Microsoft Surface tablet with Windows RT, sponsored by Microsoft Research.

Nov 20, 2012

Prof. Gail C. Murphy is a keynote speaker at MSR 2013!

Nov 05, 2012

We've announced our first Program Committees. They look forward to your research, practice, and data papers as well as challenge reports! Stay tuned for more.

Sep 12, 2012

MSR 2013 Website Launched!

Welcome to the official website of MSR 2013

The Mining Software Repositories (MSR) field analyzes the rich data available in software repositories to uncover interesting and actionable information about software systems and projects. The goal of this two-day working conference is to advance the science and practice of MSR.

The **10th Working Conference on Mining Software Repositories** is sponsored by IEEE TCSE and ACM SIGSOFT.

Accepted Papers:

Statistics:

Research/Practice (full):	81 submitted, 31 accepted	(38%)
Research/Practice (short):	22 submitted, 5 accepted	(23%)
Data:	27 submitted, 15 accepted	(56%)
Challenge:	29 submitted, 12 accepted	(41%)

Technical Track - Full Papers

- Strategies for Avoiding Text Fixture Smells during Software Evolution** *Michaela Greiler, Andy Zaidman, Arie van Deursen and Margaret-Anne Storey.*
- Happy Birthday! A trend analysis on ten years' worth of MSR papers** *Kevin Wyckmans, Alessandro Murgia, Ahmed Lamkanfi and Serge Demeyer.*
- Testing Principles, Current Practices and Effects of Change Localization** *Steven Raemaekers, Gabriela Nane, Arie van Deursen and Joost Visser.*
- Mining Succinct and High-Coverage API Usage Patterns from Source Code** *Jue Wang, Yingnong Dang, Hongyu Zhang, Kai Chen, Tao Xie and Dongmei Zhang.*
- Using Citation Influence to Predict Software Defects** *Wei Hu and Kenny Wong.*
- Assisting Code Search with Automatic Query Reformulation for Bug Localization** *Bunyamin Sisman and Avinash Kak.*

Mining Software Repositories

```
mozilla_log_snippet.txt
1 RCS file: /cvsroot/mozilla/layout/generic/nsViewportFrame.cpp,v-
2 Working file: layout/generic/nsViewportFrame.cpp-
3 head: 1.75-
4 branch:-
5 locks: strict-
6 access list:-
7 symbolic names:-
8 > DOM_AGNOSTIC_BRANCH: 1.75.0.16-
9 > DOM_AGNOSTIC_BASE: 1.75-
10 > MOZILLA_1_8_BRANCH: 1.75.0.14-
11 > MOZILLA_1_8_BASE: 1.75-
12 > REFLOW_20050804_BRANCH: 1.75.0.12-
13 > REFLOW_20050804_BASE: 1.75-
14 > SPLITWINDOW_20050714_INITIAL_TRUNK_LANDING: 1.75-
15 > SPLITWINDOW_20050714_BRANCH: 1.75.0.10-
16 > SPLITWINDOW_20050714_BASE: 1.75-
17 > MOZILLA_1_8b3_RELEASE: 1.75-
18 > THUNDERBIRD_1_1a2_RELEASE: 1.75-
19 > FIREFOX_1_1a2_RELEASE: 1.75-
20 > THUNDERBIRD_1_1a1_RELEASE: 1.75-
21 > FIREFOX_1_1a1_RELEASE: 1.75-
22 > BSMEDBERG_SECURITY_PLAYGROUND_20050512_BRANCH: 1.75.0.8-
23 > BSMEDBERG_SECURITY_PLAYGROUND_20050512_BASE: 1.75-
24 > REFLOW_20050429_BRANCH: 1.75.0.6-
25 > REFLOW_20050429_BASE: 1.75-
26 > SOFTWARE_UPDATE_20050428_BRANCH: 1.75.0.4-
27 > SOFTWARE_UPDATE_20050428_BASE: 1.75-
28 > REFLOW_20050315_BRANCH: 1.75.0.2-
29 > REFLOW_20050315_BASE: 1.75-
30 > MOZILLA_1_8b1_RELEASE: 1.75-
31 > WEBFORMS_20050202_BRANCH: 1.73.0.10-
32 > WEBFORMS_20050202_BASE: 1.73-
33 > PREFERENCES_20050201_BRANCH: 1.73.0.8-
34 > PREFERENCES_20050201_BASE: 1.73-
35 > MOZILLA_1_8a6_RELEASE: 1.73-
36 > REFLOW_20050111_BRANCH: 1.73.0.6-
37 > REFLOW_20050111_BASE: 1.73-
38 > COMMANDLINES_20050109_BRANCH: 1.73.0.4-
39 > COMMANDLINES_20050109_BASE: 1.73-
40 > XFORMS_20050106_BRANCH: 1.73.0.2-
41 > XFORMS_20050106_BASE: 1.73-
42 > PREFERENCES_20050101_BRANCH: 1.72.0.4-
43 > PREFERENCES_20050101_BASE: 1.72-
44 > REFLOW_20041213_BRANCH: 1.72.0.2-
45 > REFLOW_20041213_BASE: 1.72-
46 keyword substitution: kv-
47 total revisions: 85;+ selected revisions: 85-
48 description:-
49 -----
50 revision 1.75-
51 date: 2005/02/07 02:02:47; author: bzbarsky@mit.edu; state: Exp; lines: +0 -0-
52 branches: 1.75.2; 1.75.6; 1.75.12;-
53 Previous checkin was for bug 244581-
54 -----
55 revision 1.74-
56 date: 2005/02/07 01:58:24; author: bzbarsky@mit.edu; state: Exp; lines: +6 -14-
57 Remove prescontext args for some nsIFrame methods-
58 (Append/Insert/Remove/ReplaceFrames) and for some methods in table land. Patch-
59 by Vidar Brant Haarr <bugmail@qin.org>, r+s=bzbarsky-
60 -----
Line: 4 Column: 8 Plain Text Tab Size: 4
```

```
mozilla_log_snippet.txt
61 revision 1.73-
62 date: 2005/01/01 17:26:27; author: neil@parkwaycc.co.uk; state: Exp; lines: +2 -3;-
63 branches: 1.73.6;-
64 Bug 276100 GetOriginToView has unused PresContext argument p=bugmail@qin.org r=sr=dbaron-
65 -----
66 revision 1.72-
67 date: 2004/12/08 19:39:49; author: dbaron@dbaron.org; state: Exp; lines: +0 -0-
68 Bug 272151: reorganize files within layout, r=sr=roc: Empty checkin to files that were repository-copied. Please see bug 272151 for old locations and see cvs history at previous
69 locations for history on branches.-
70 -----
71 revision 1.71-
72 date: 2004/11/14 20:22:58; author: roc@cs.cmu.edu; state: Exp; lines: +7 -13-
73 Bug 261064. Compute static absolute positions more accurately. Also, ensure that static absolutely positioned frames are moved when their placeholders move. Also, ensure that
74 absolutely positioned frames positioned relative to the left or bottom edge of their container move when the container size changes. r=sr=dbaron (rubberstamp)-
75 -----
76 revision 1.70-
77 date: 2004/07/31 23:15:10; author: bryner@brianryner.com; state: Exp; lines: +8 -8-
78 Change nsIPresContext to nsPresContext globally, follow-up to bug 253470. rs=roc-
79 -----
80 revision 1.69-
81 date: 2004/04/18 14:30:24; author: gerv@gerv.net; state: Exp; lines: +10 -10-
82 Bug 236613: change to MPL/LGPL/GPL tri-license.-
83 -----
84 revision 1.68-
85 date: 2004/03/12 04:40:16; author: dbaron@dbaron.org; state: Exp; lines: +3 -4-
86 DeCOMify nsIFrame::IsPercentageBase and rename to IsContainingBlock. b=237169 r=sr=bzbarsky a=brendan-
87 -----
88 revision 1.67-
89 date: 2004/03/10 03:09:03; author: roc@cs.cmu.edu; state: Exp; lines: +1 -3-
90 Bug 235264. Clean up invalidation to go through a single nsIFrame::Invalidate function and take account of 'outline' where necessary. r=sr=dbaron-
91 -----
92 revision 1.66-
93 date: 2004/01/09 19:21:20; author: roc@cs.cmu.edu; state: Exp; lines: +18 -15-
94 Bug 210269. Reduce use of SetAttr during reflow. Also, fix up the API that gets the scrollbar dimensions of a scrollframe to always work right for RTL. r=sr=bryner-
95 -----
96 revision 1.65-
97 date: 2004/01/09 14:20:50; author: roc@cs.cmu.edu; state: Exp; lines: +11 -20-
98 Bug 190735. deCOMtinate nsIFrame::FirstChild and nsIFrame::GetAdditionalChildListName. r=sr=bz-
99 -----
100 revision 1.64-
101 date: 2003/10/31 20:19:11; author: uid502; state: Exp; lines: +3 -6-
102 Bug 190735. DeCOMtinate nsIFrame::GetFrameType. r=sr=bzbarsky-
103 -----
104 revision 1.63-
105 date: 2003/02/22 20:13:12; author: bzbarsky@mit.edu; state: Exp; lines: +39 -47-
106 Reflow main content before reflowing fixed-pos frames so that the placeholders-
107 are in the right places. Bug 90270 and dependencies. r=sr=roc+moz-
108 -----
109 revision 1.62-
110 date: 2003/02/22 15:34:03; author: dbaron@dbaron.org; state: Exp; lines: +0 -10-
111 Remove nsISizeOfHandler and associated SizeOf methods. b=106792 r=bzbarsky sr=jst-
112 -----
113 revision 1.61-
114 date: 2003/02/13 19:10:50; author: bzbarsky@mit.edu; state: Exp; lines: +0 -1-
115 Remove leftover reflow reason hack in fixed-pos reflow. Bug 192291,-
116 r=sr=dbaron, a=asa-
117 -----
118 revision 1.60-
119 date: 2003/01/20 18:04:29; author: karnaze@netscape.com; state: Exp; lines: +70 -351-
120 bug 179683 - implement printing of fixed positioned elements, clean up ViewportFrame. sr=roc, r=bzbarsky.-
```

Line: 4 Column: 8

Plain Text

Tab Size: 4

[Bug 377413 - Dynamic delet](#)

[https://bugzilla.mozilla.org/show_bug.cgi?id=377413](#)

Bugzilla@Mozilla - Bug 377413 Dynamic delete of form elements leaves references sometimes Last modified: 2008-09-06 19:36:36 PDT

[Home](#) | [New](#) | [Browse](#) | [Search](#) | (?) | [Reports](#) | [Requests](#) | [Help](#) | [New Account](#) | [Log In](#) | [Forgot Password](#)

First Last Prev Next This bug is not in your last search results.

Bug 377413 - Dynamic delete of form elements leaves references sometimes [Last Comment](#)

Status:	UNCONFIRMED	Reported:	2007-04-13 10:11 PDT by Les Barstow
Whiteboard:		Modified:	2008-09-06 19:36 PDT (History)
Keywords:		CC List:	5 users (show)
Product:	Core	See Also:	
Component:	DOM: Core & HTML	tracking-firefox5:	---
Version:	Trunk	status-firefox5:	---
Platform:	All All	tracking-firefox6:	---
Importance:	-- normal with 1 vote (vote)	tracking-fennec:	---
Target Milestone:	---	blocking2.0:	---
Assigned To:	general	status2.0:	---
QA Contact:	general	blocking1.9.2:	---
URL:	http://pt6demo.profitool.com/testaddr...	status1.9.2:	---
Depends on:		blocking1.9.1:	---
Blocks:		status1.9.1:	---

[Show dependency tree / graph](#)

Attachments

[Add an attachment](#) (proposed patch, testcase, etc.)

[Summon comment box](#)

Les Barstow **2007-04-13 10:11:18 PDT** [Description](#)

```

User-Agent:      Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.3) Gecko/20070309
Firefox/2.0.0.3
Build Identifier: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.3) Gecko/20070309
Firefox/2.0.0.3

The URL above is a somewhat simple application that inserts and deletes rows in a table using
JavaScript.  Rows all contain form input fields.  Any reference to an existing form element -
statically created, or dynamically created using innerHTML or document.createElement("input") -
causes the element to become somewhat indelible.  Deleting the table row containing the input field
(or even deleting the input element itself and then deleting the row) leaves a residual reference
for any element that had been referenced while it existed.

The referenced URL document contains several sample tests that can be run.  URL contains all
necessary JavaScript and HTML.

Reproducible: Always
  
```

Metrics

Metrics are a mapping of a particular characteristic of a measured entity to a numerical value.

They can be used to quantify aspects of quality.

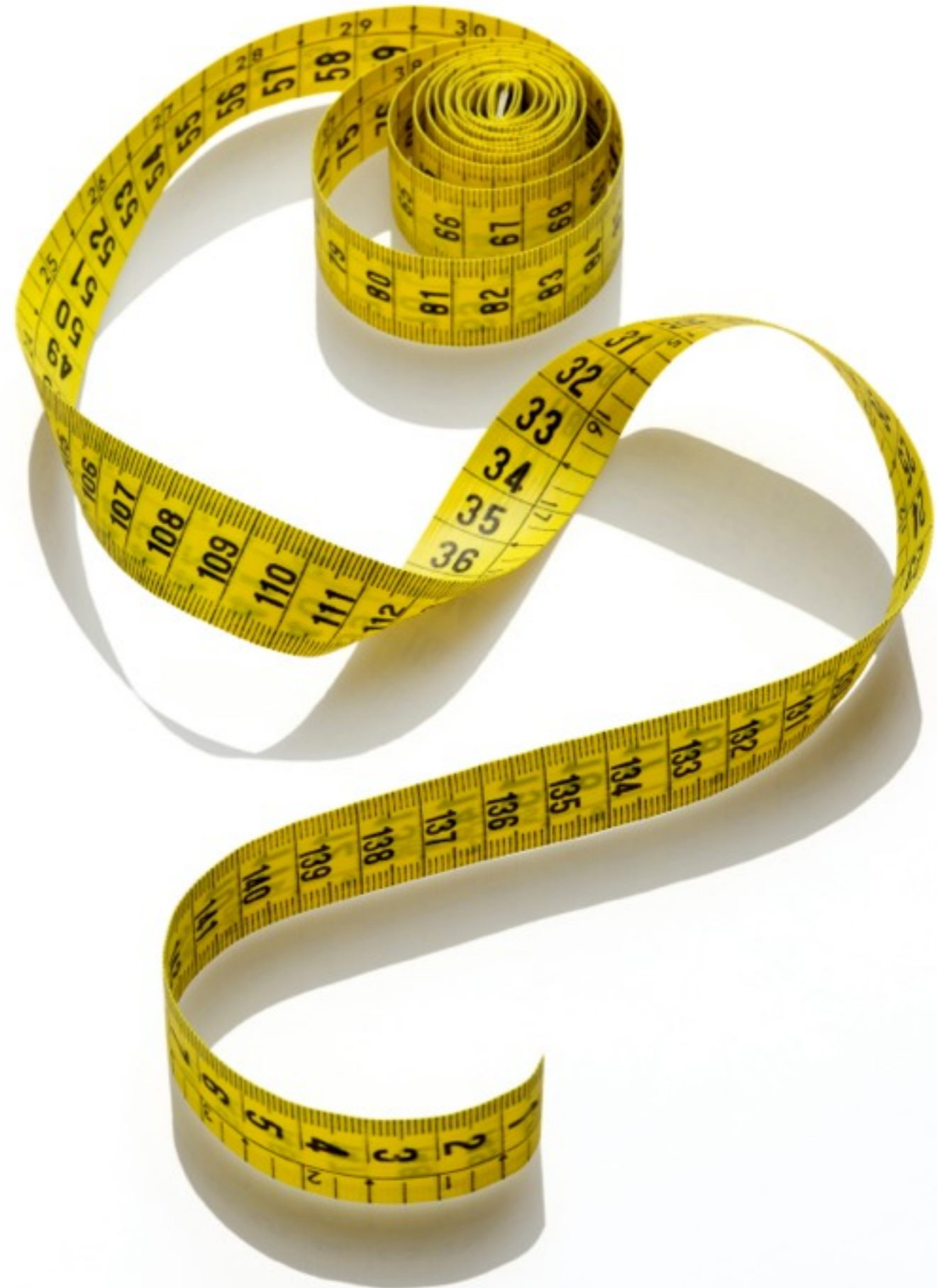
Measuring software is cheap and can be automated.

Numbers are just numbers; don't trust them.

They capture symptoms, not causes.

Hard for developers to deal with them.

Inflation of measurements



Yesterday I met a system...

Q: How many lines of code?

A: 35'000 LOC

*Q: How many functions/
methods?*

A: 3600 NOM

Q: How many classes?

A: 380 NOC

Is it “normal” to have such a system?

What about coupling or cohesion?



Yesterday I met a system...

Q: How many lines of code?

A: 35'000 LOC

*Q: How many functions/
methods?*

A: 3600 NOM

Q: How many classes?

A: 380 NOC

Is it “normal” to have such a system?

What about coupling or cohesion?



The Overview Pyramid

A metrics-based means to both describe and characterize the structure of an object-oriented system by quantifying its **complexity, coupling, and usage of inheritance**

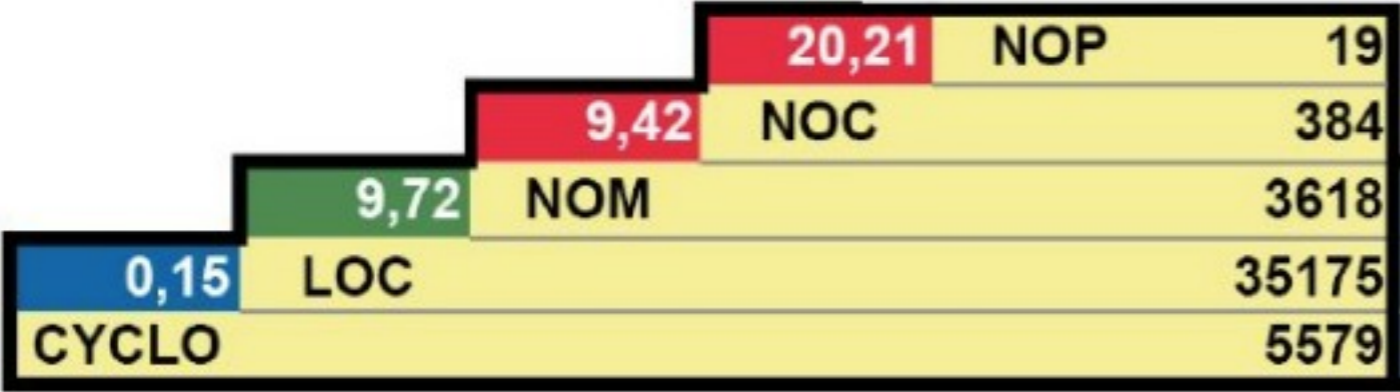
Measuring these three aspects at system level provides a comprehensive **characterization** of an entire system.



The left side: System Size & Complexity

Direct metrics: NOP, NOC, LOC, CYCLO

Derived Metrics: NOC/P, NOM/C, LOC/M, Cyclo/LOC



The Overview Pyramid in Detail - Object Oriented Metrics in Practice, Lanza and Marinescu.

The left side: System Size & Complexity

Direct metrics: NOP, NOC, LOC, CYCLO

Derived Metrics: NOC/P, NOM/C, LOC/M, Cyclo/LOC

The right side: System Coupling

Direct metrics: CALLS, FANOUT

Derived Metrics: CALLS/M, FANOUT/CALL



The Overview Pyramid in Detail - Object Oriented Metrics in Practice,
Lanza and Marinescu.

The left side: System Size & Complexity

Direct metrics: NOP, NOC, LOC, CYCLO

Derived Metrics: NOC/P, NOM/C, LOC/M, Cyclo/LOC

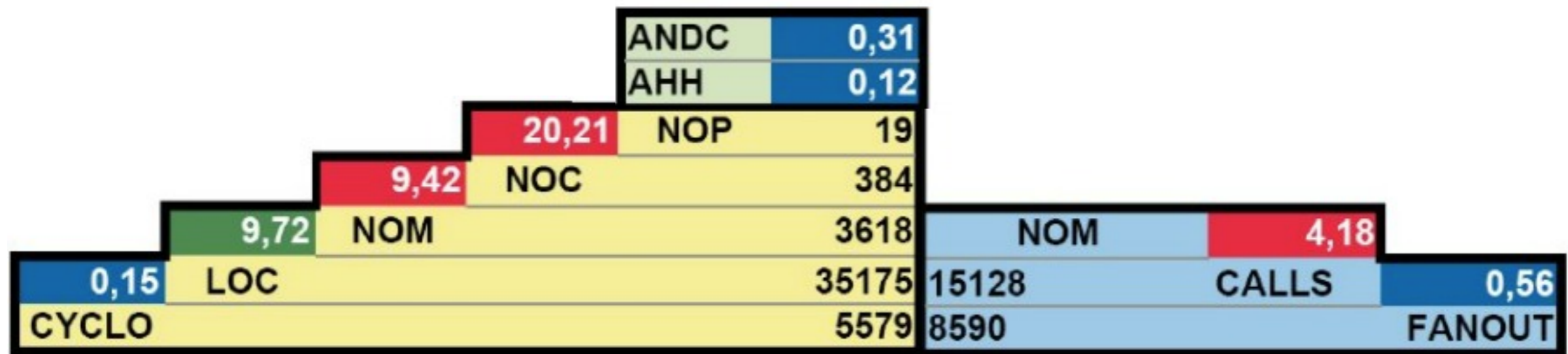
The right side: System Coupling

Direct metrics: CALLS, FANOUT

Derived Metrics: CALLS/M, FANOUT/CALL

The top: System Inheritance

Direct metrics: ANDC, AHH



The Overview Pyramid in Detail - Object Oriented Metrics in Practice, Lanza and Marinescu.

The left side: System Size & Complexity

Direct metrics: NOP, NOC, LOC, CYCLO

Derived Metrics: NOC/P, NOM/C, LOC/M, Cyclo/LOC

The right side: System Coupling

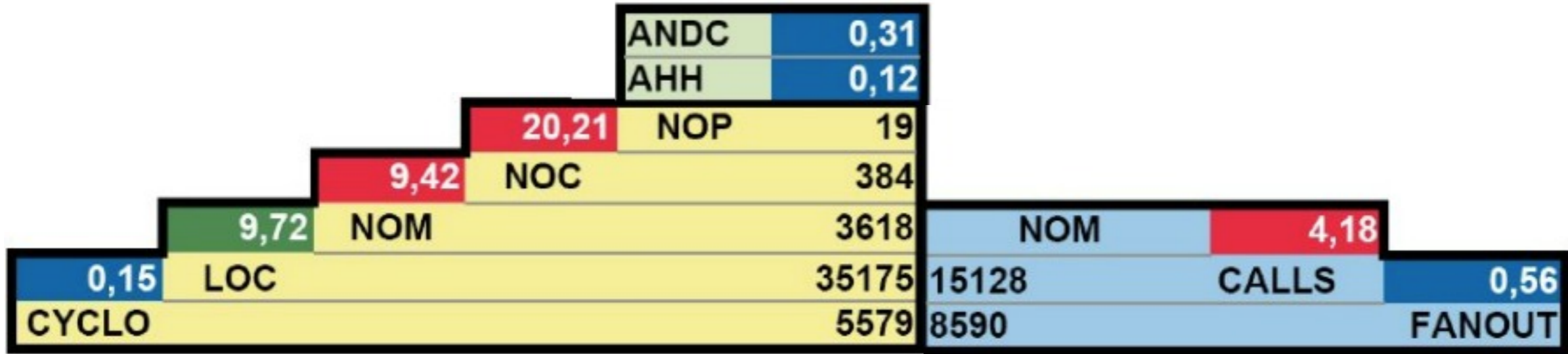
Direct metrics: CALLS, FANOUT

Derived Metrics: CALLS/M, FANOUT/CALL

The top: System Inheritance

Direct metrics: ANDC, AHH

What about reference points?



The Overview Pyramid in Detail - Object Oriented Metrics in Practice, Lanza and Marinescu.

The left side: System Size & Complexity

Direct metrics: NOP, NOC, LOC, CYCLO
 Derived Metrics: NOC/P, NOM/C, LOC/M, Cyclo/LOC

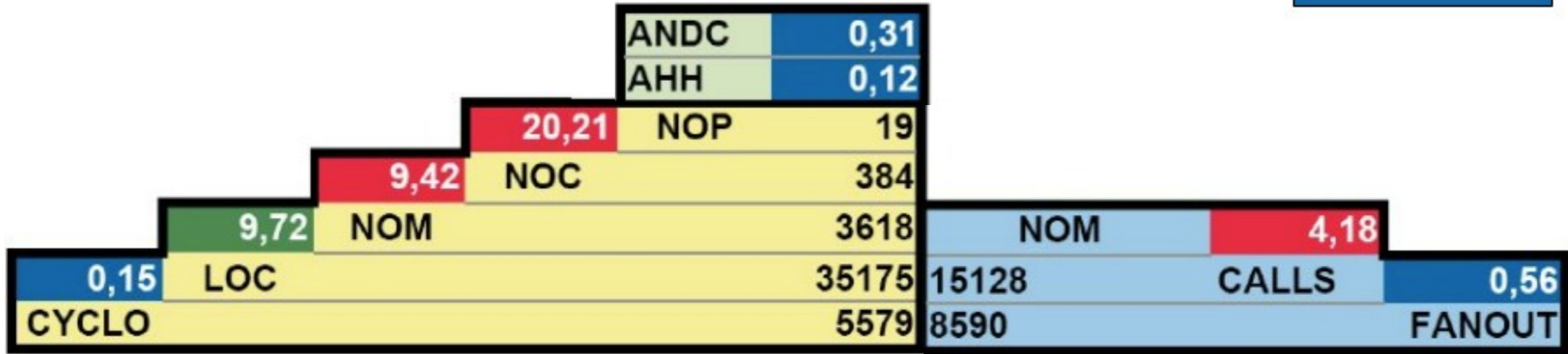
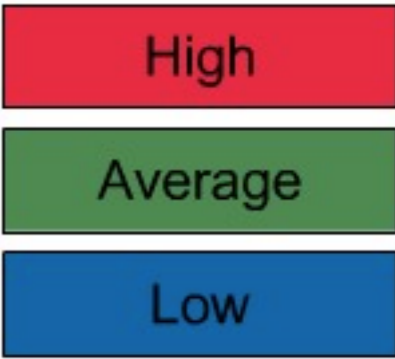
The right side: System Coupling

Direct metrics: CALLS, FANOUT
 Derived Metrics: CALLS/M, FANOUT/CALL

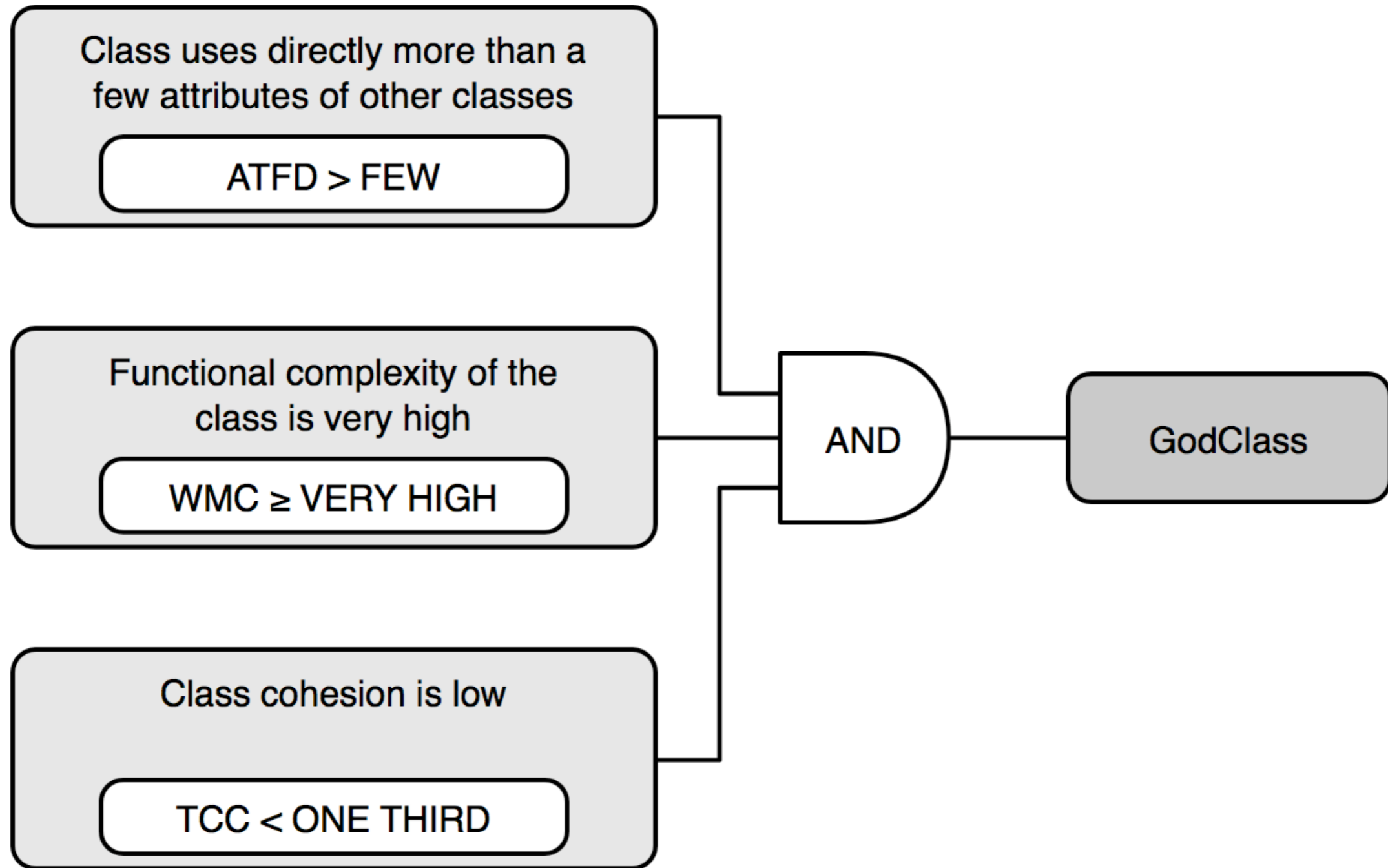
The top: System Inheritance

Direct metrics: ANDC, AHH

What about reference points?



The Overview Pyramid in Detail - Object Oriented Metrics in Practice, Lanza and Marinescu.



Identity Disharmony: God Class

Class uses directly more than a few attributes of other classes

ATFD > FEW

Functional complexity of the class is very high

WMC ≥ VERY HIGH

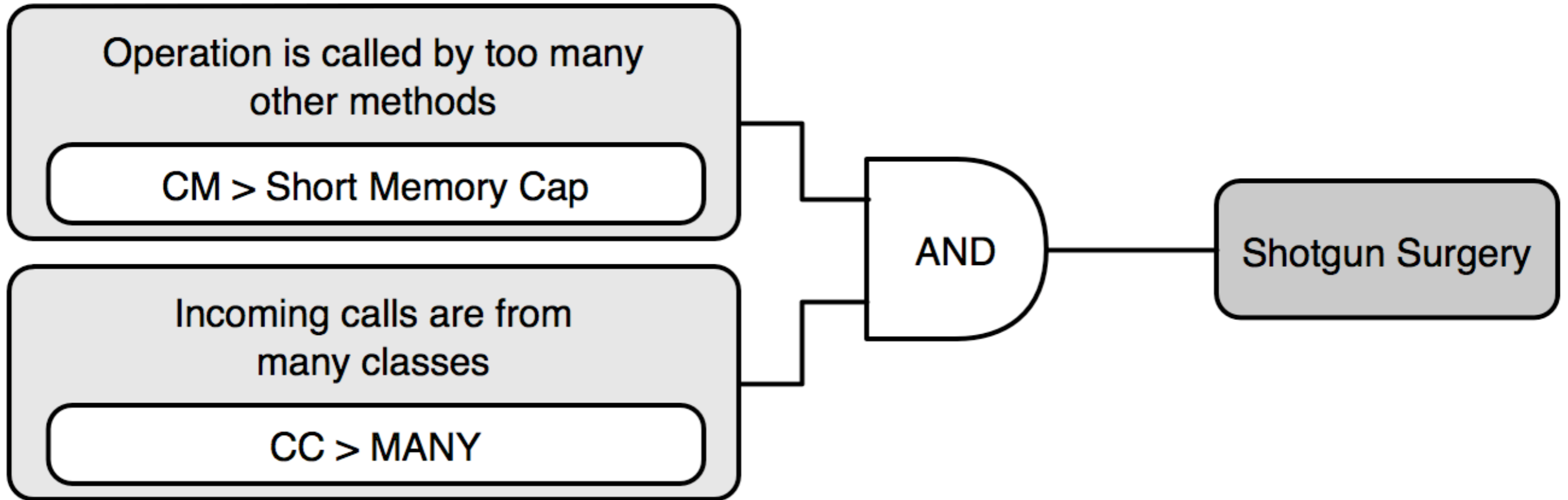
Class cohesion is low

TCC < ONE THIRD

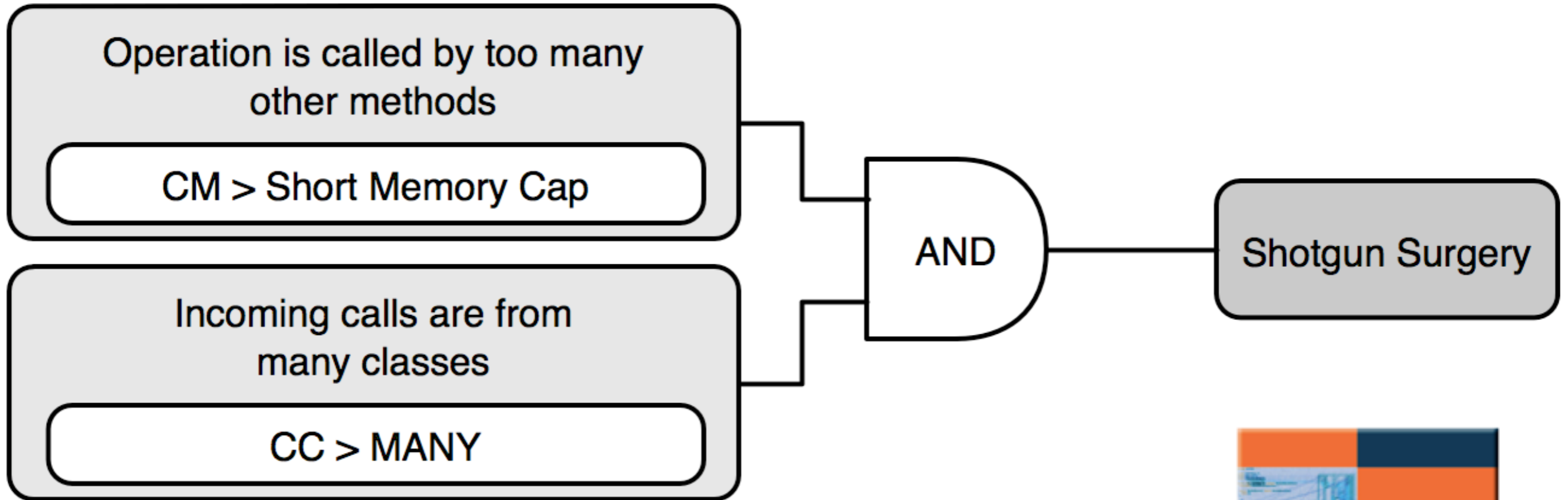
AND

GodClass

What about reference points?



Collaboration Disharmony: Shotgun Surgery



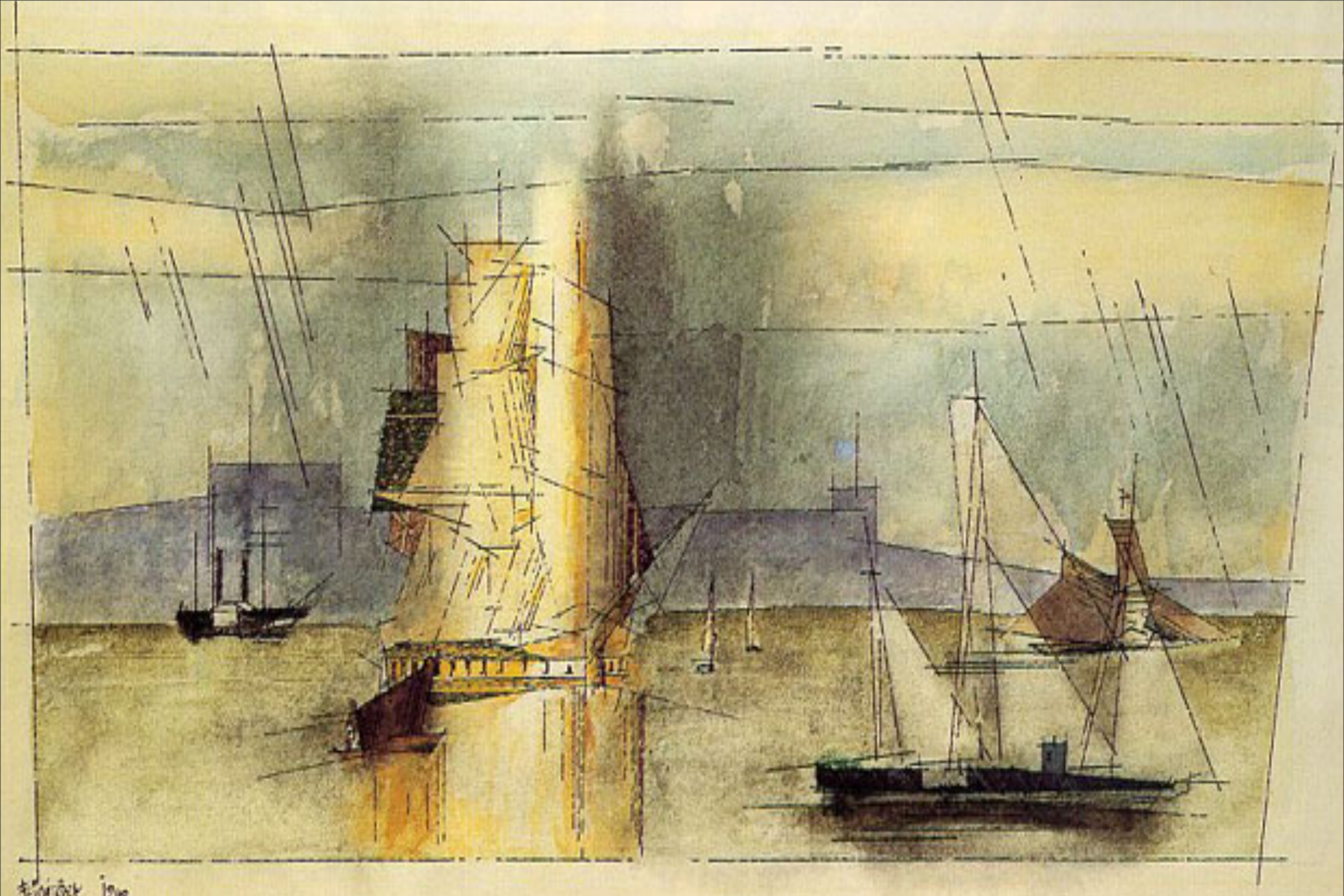
Collaboration Disharmony: Shotgun Surgery



What is the meaning of those numbers?



What is the meaning of those numbers?

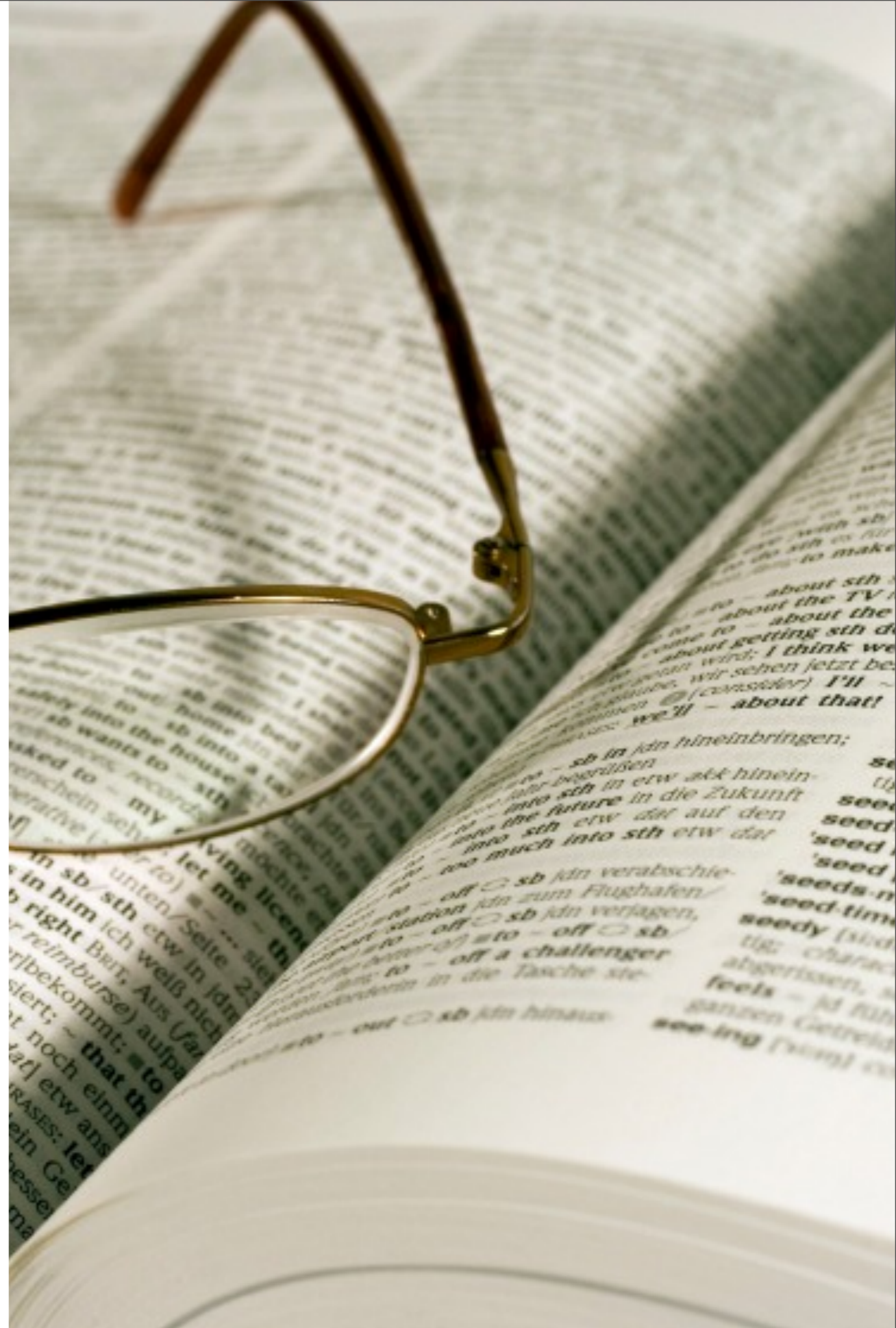


Can you understand the beauty of a painting by measuring the size of its frame, or by counting the number of colors the artist used, etc?

Software Visualization

The use of the crafts of typography, graphic design, animation, and cinematography with modern human computer interaction and computer graphics technology to facilitate both the human understanding and effective use of computer software.

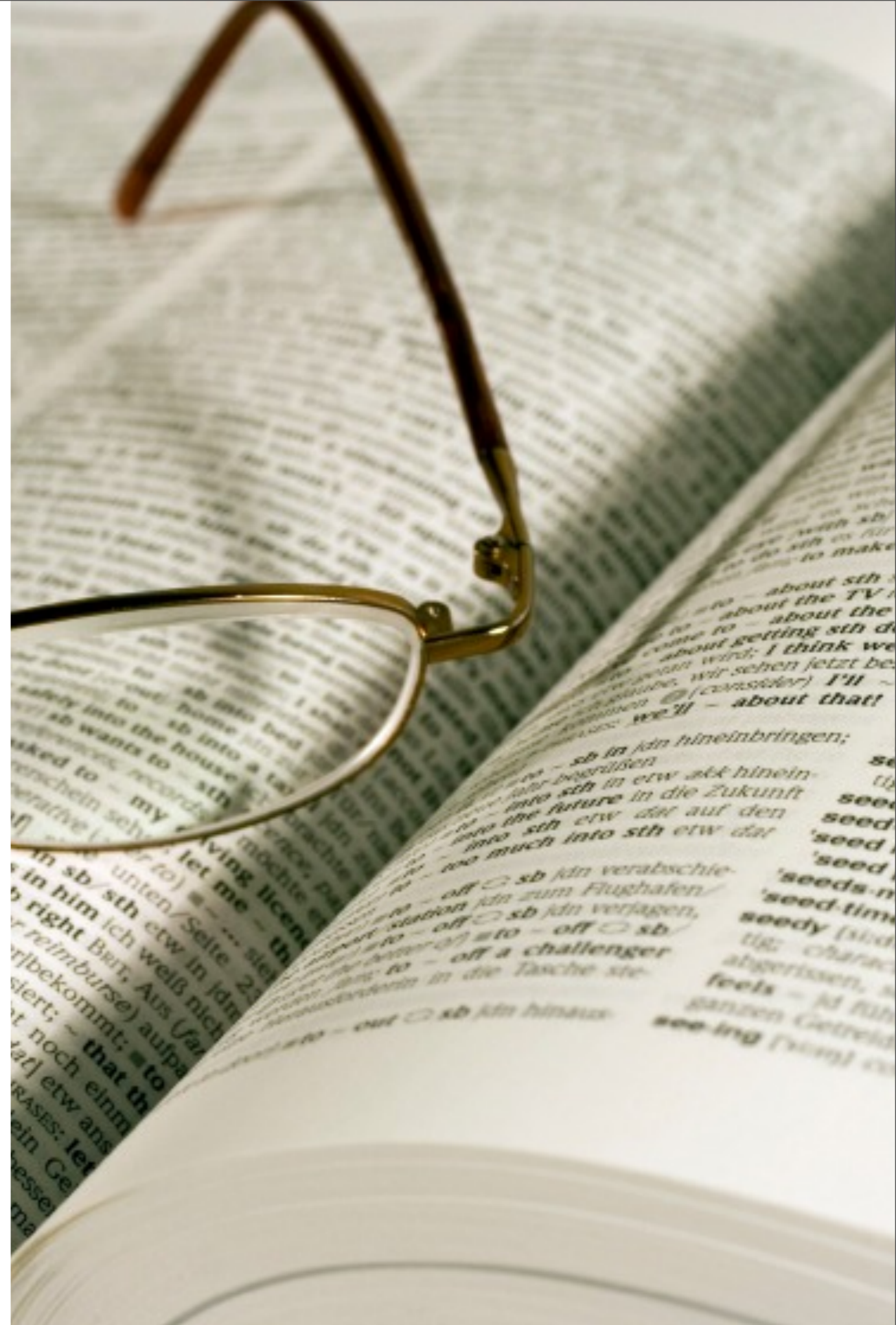
Stasko, 1998



Software Visualization

The use of the crafts of typography, graphic design, animation, and cinematography with modern human computer interaction and computer graphics technology to facilitate both the human understanding and effective use of computer software.

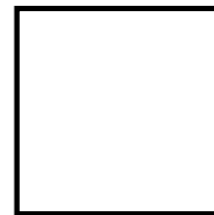
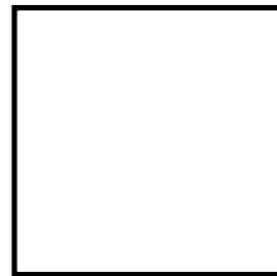
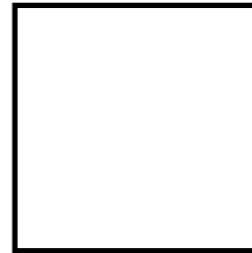
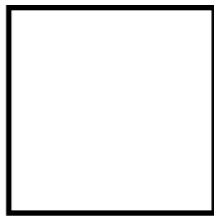
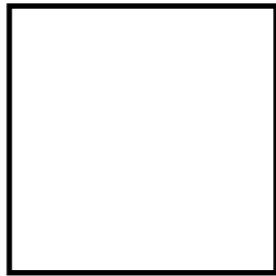
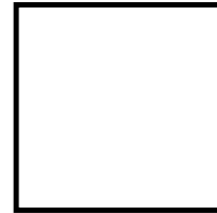
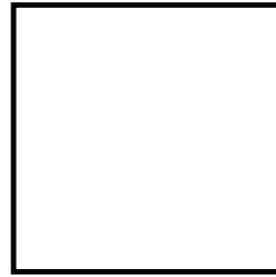
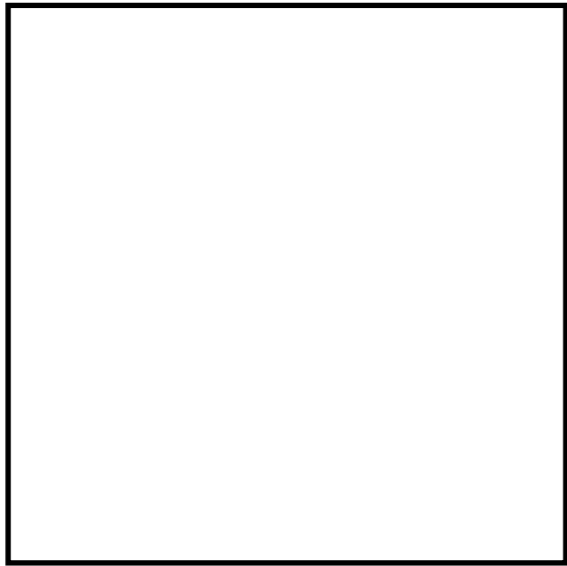
Lanza, 2010



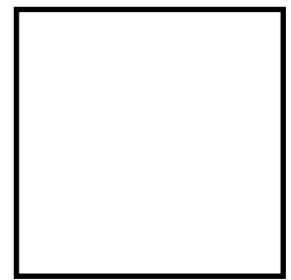
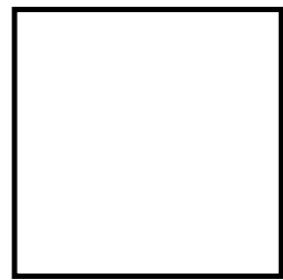
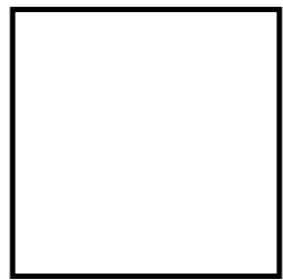
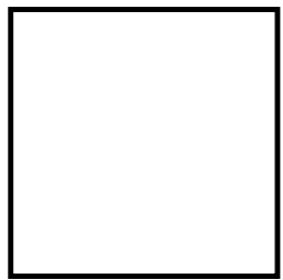
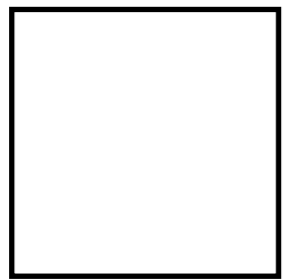
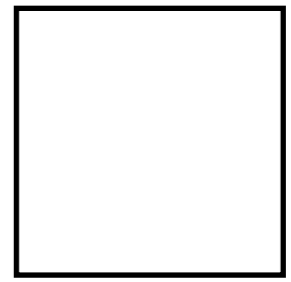
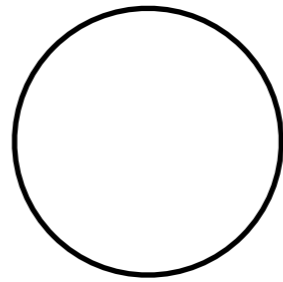
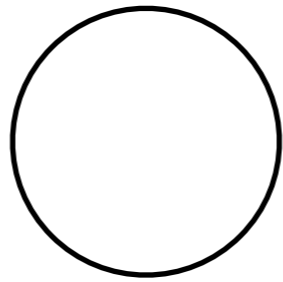
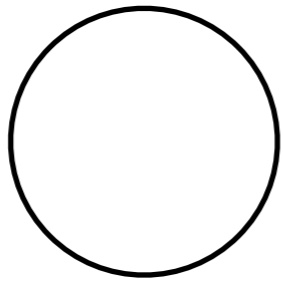
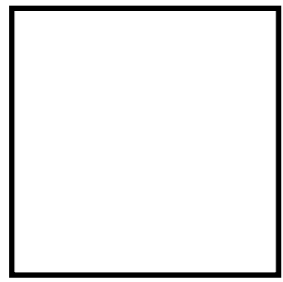
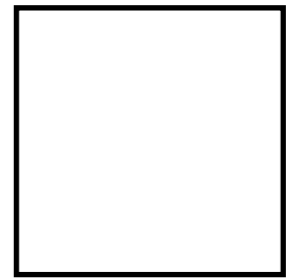
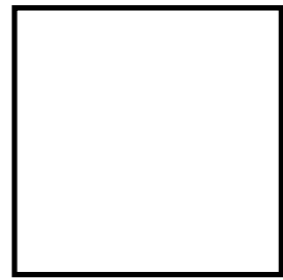
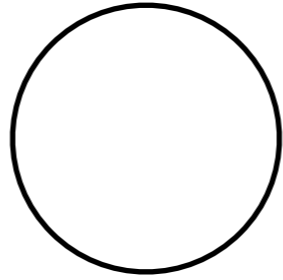
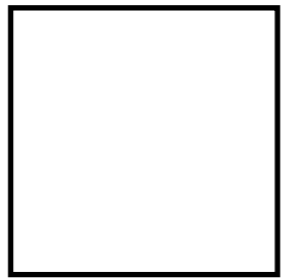
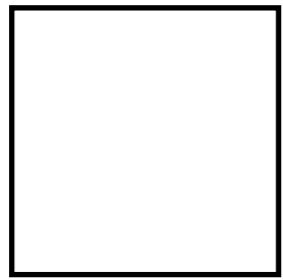
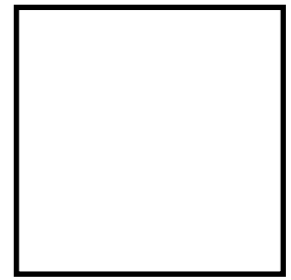
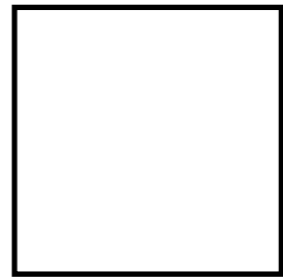
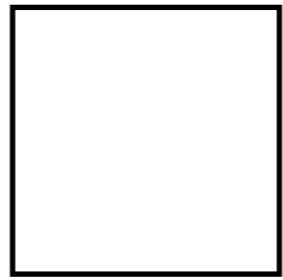
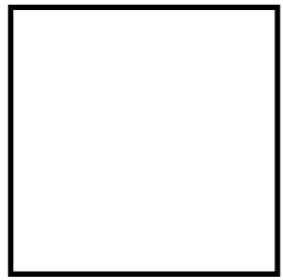
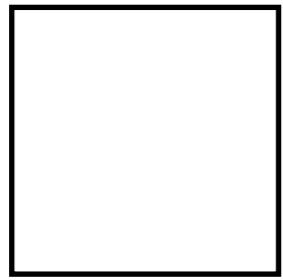
Few's Gestalt Principles

1. Proximity
2. Similarity
3. Closure
4. Enclosure
5. Continuity
6. Connection

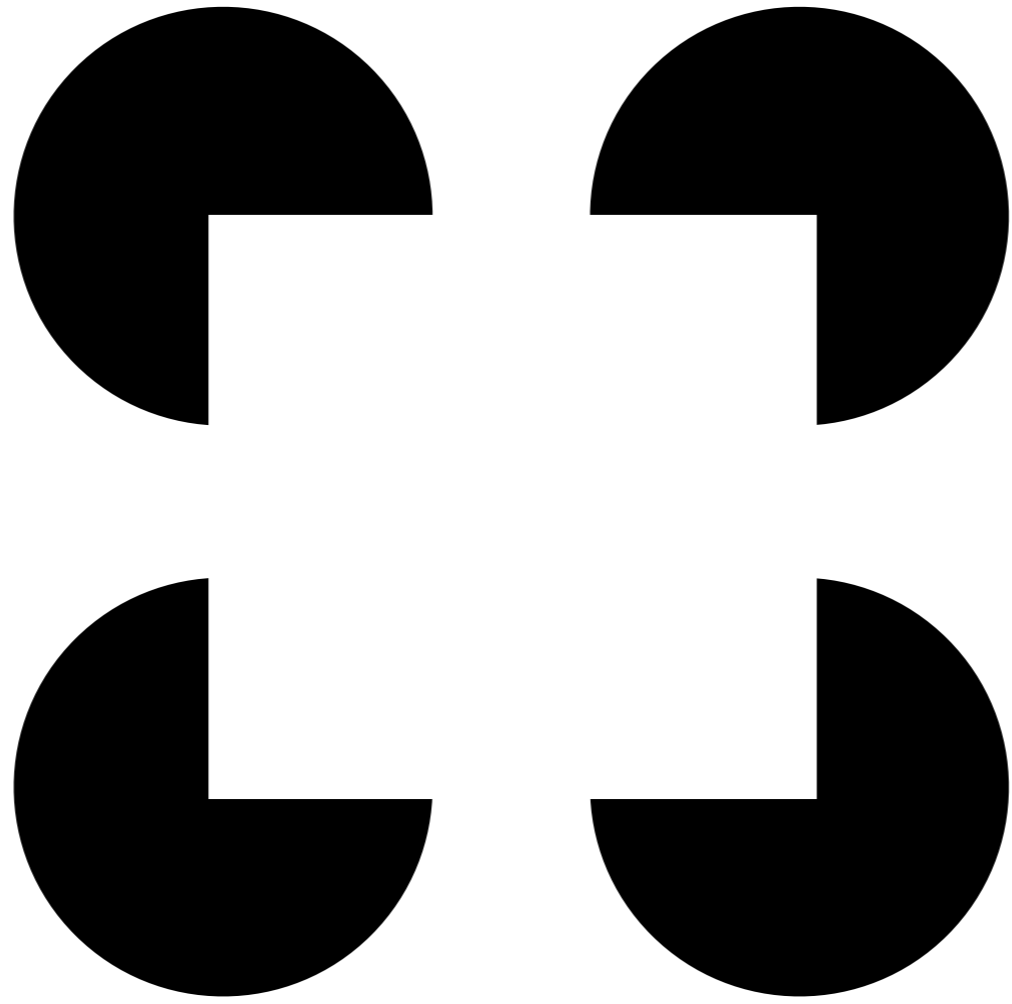




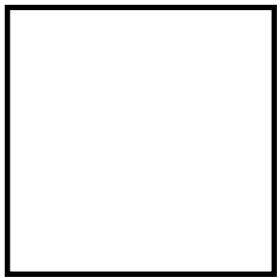
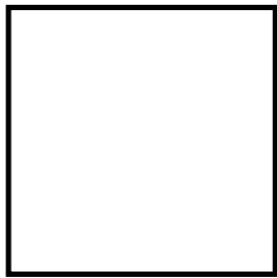
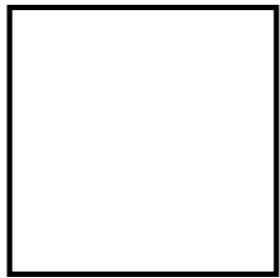
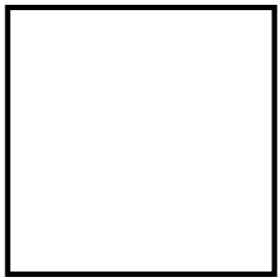
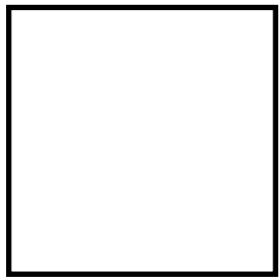
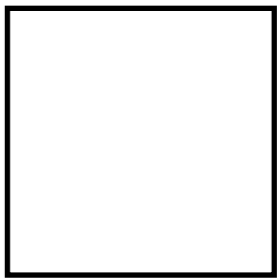
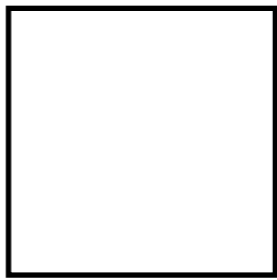
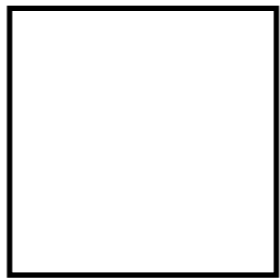
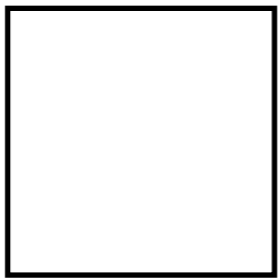
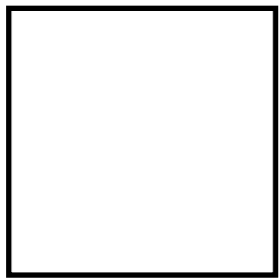
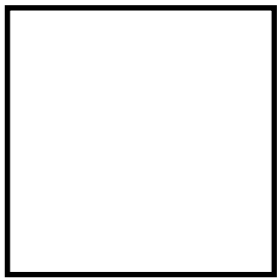
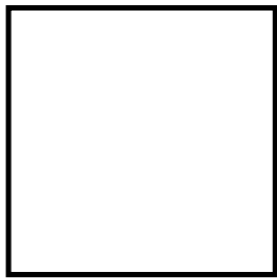
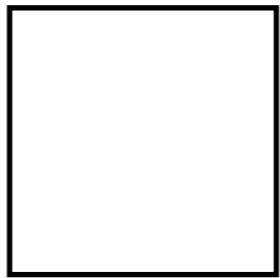
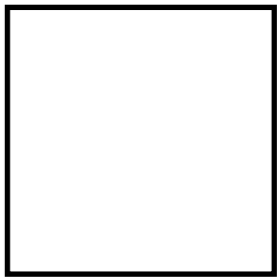
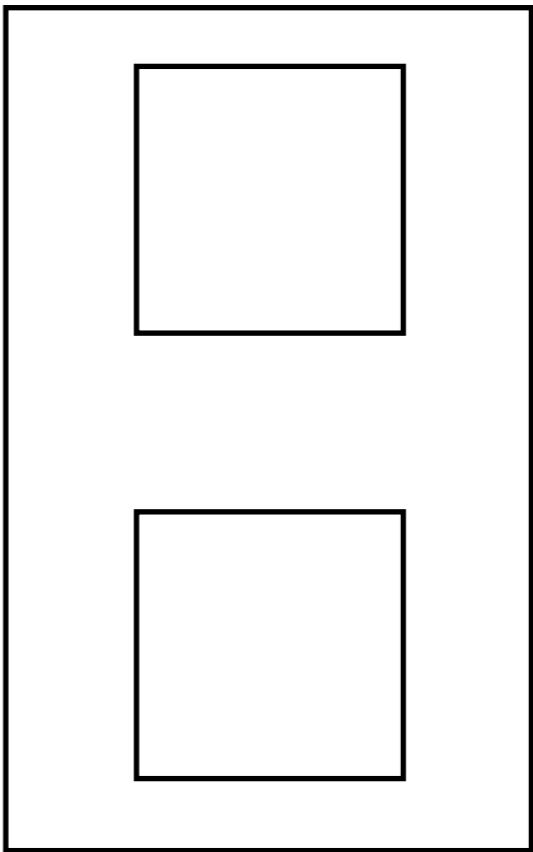
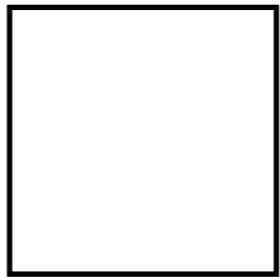
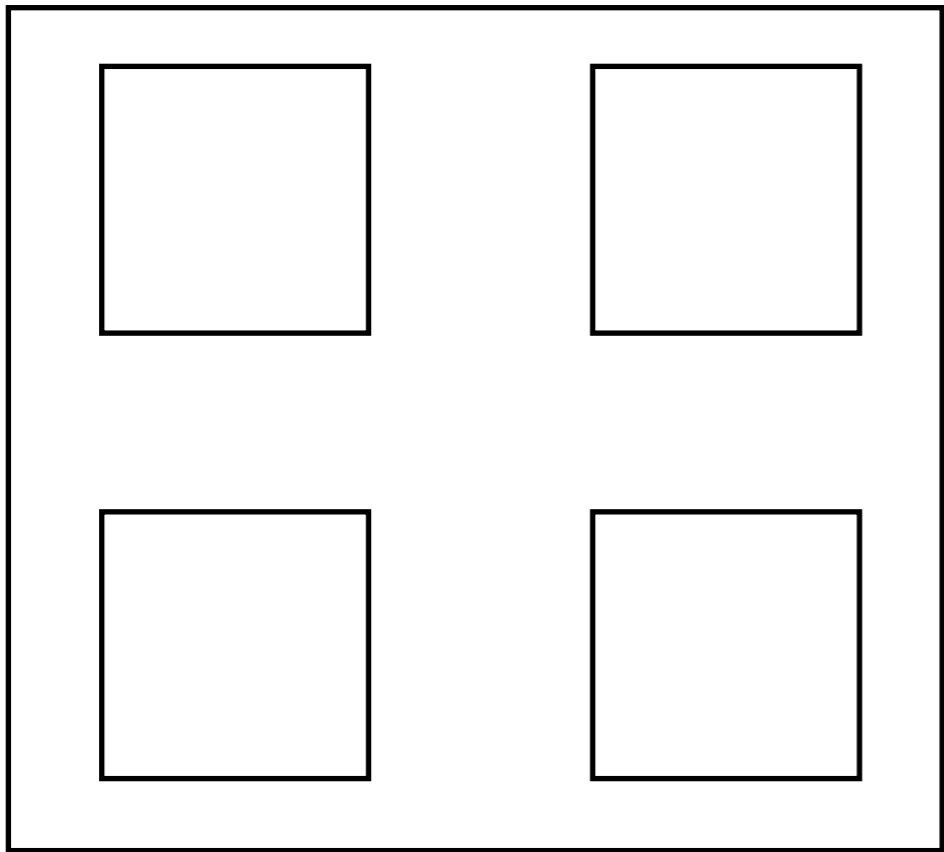
Principle of Proximity



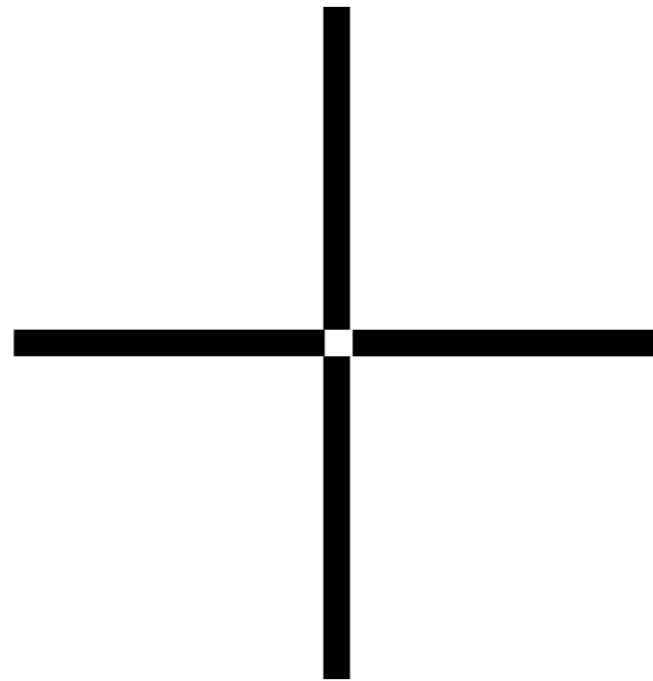
Principle of Similarity



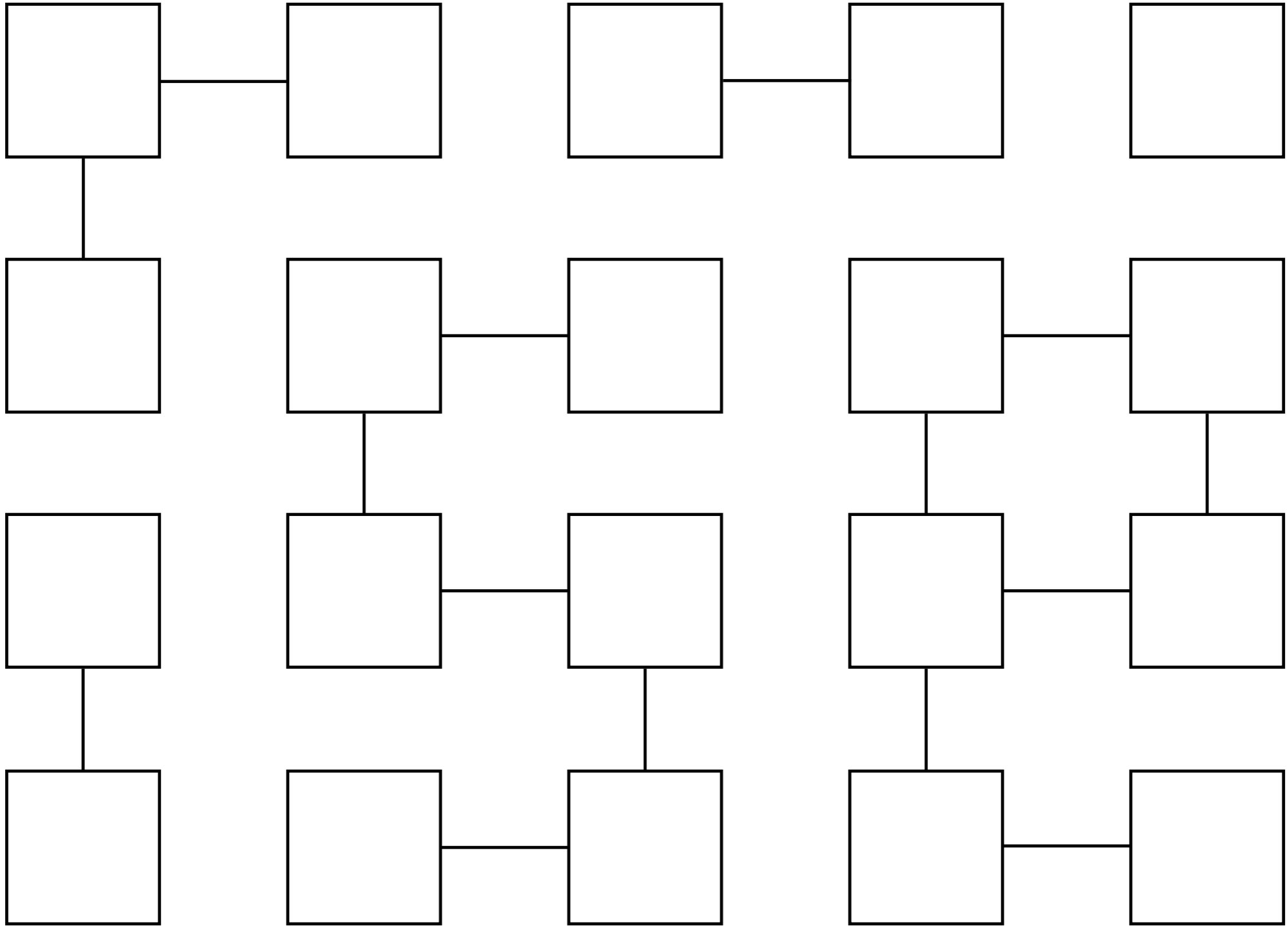
Principle of Closure



Principle of Enclosure



Principle of Continuity



Principle of Connection

Tufte's 6 Information Visualization Principles (1)

1. Tell the truth

Clear and thorough labeling, avoid distortion

2. Show the data

Reduce unnecessary non-data information to a bare minimum

3. Present many numbers in small space

Use Polymetric Views, but use them wisely



Tufte's 6 Information Visualization Principles (1)

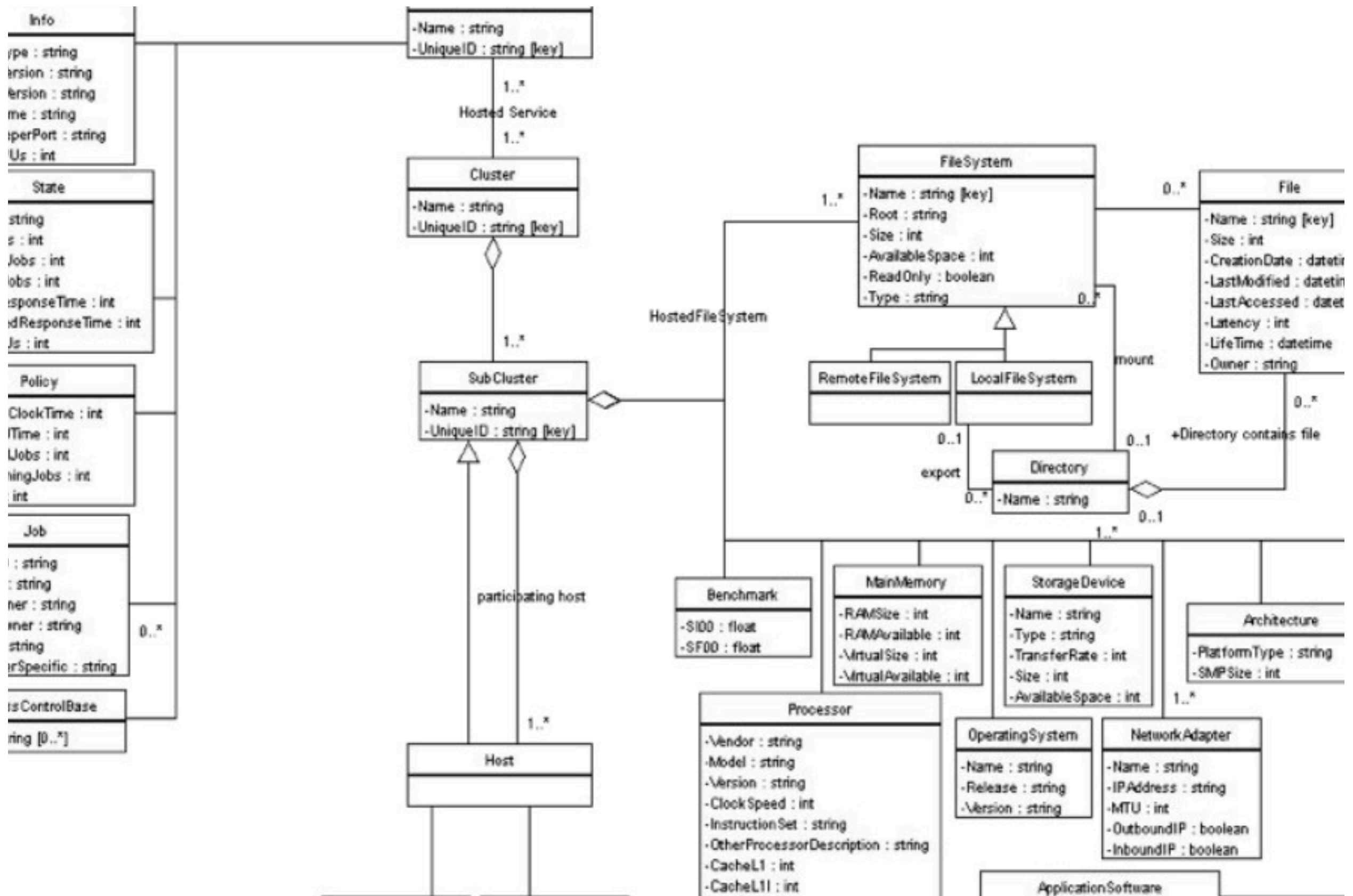
4. Help the viewer **think about the information** rather than about the methodology and the design
Remove redundant data and unnecessary design elements
5. Encourage the eye to **compare the data**
Contrast different pieces of data
6. **Make large data sets coherent**
Use three viewing depths: the overall structure, detail information and the implicit context



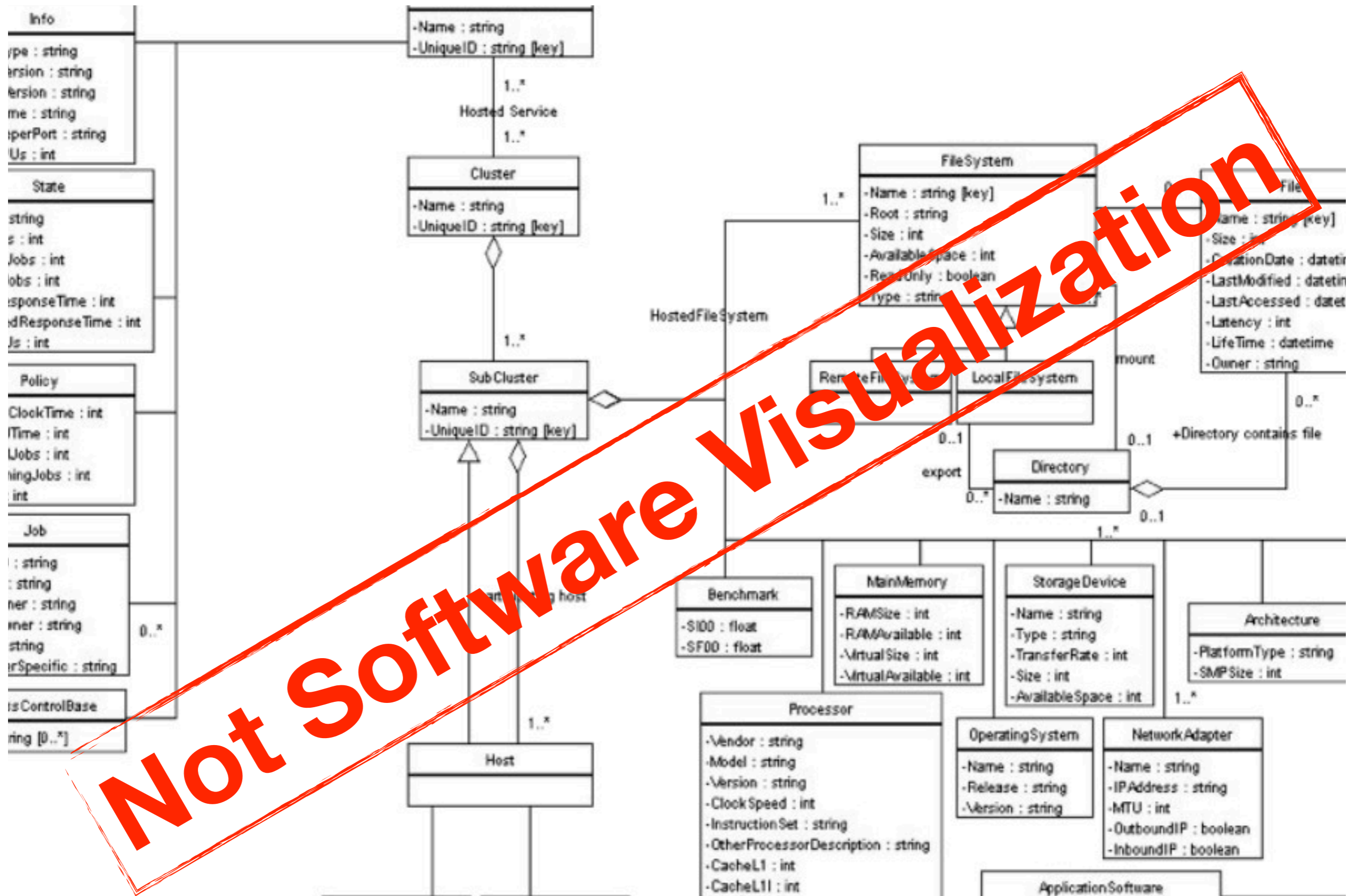


About Nodes and Edges

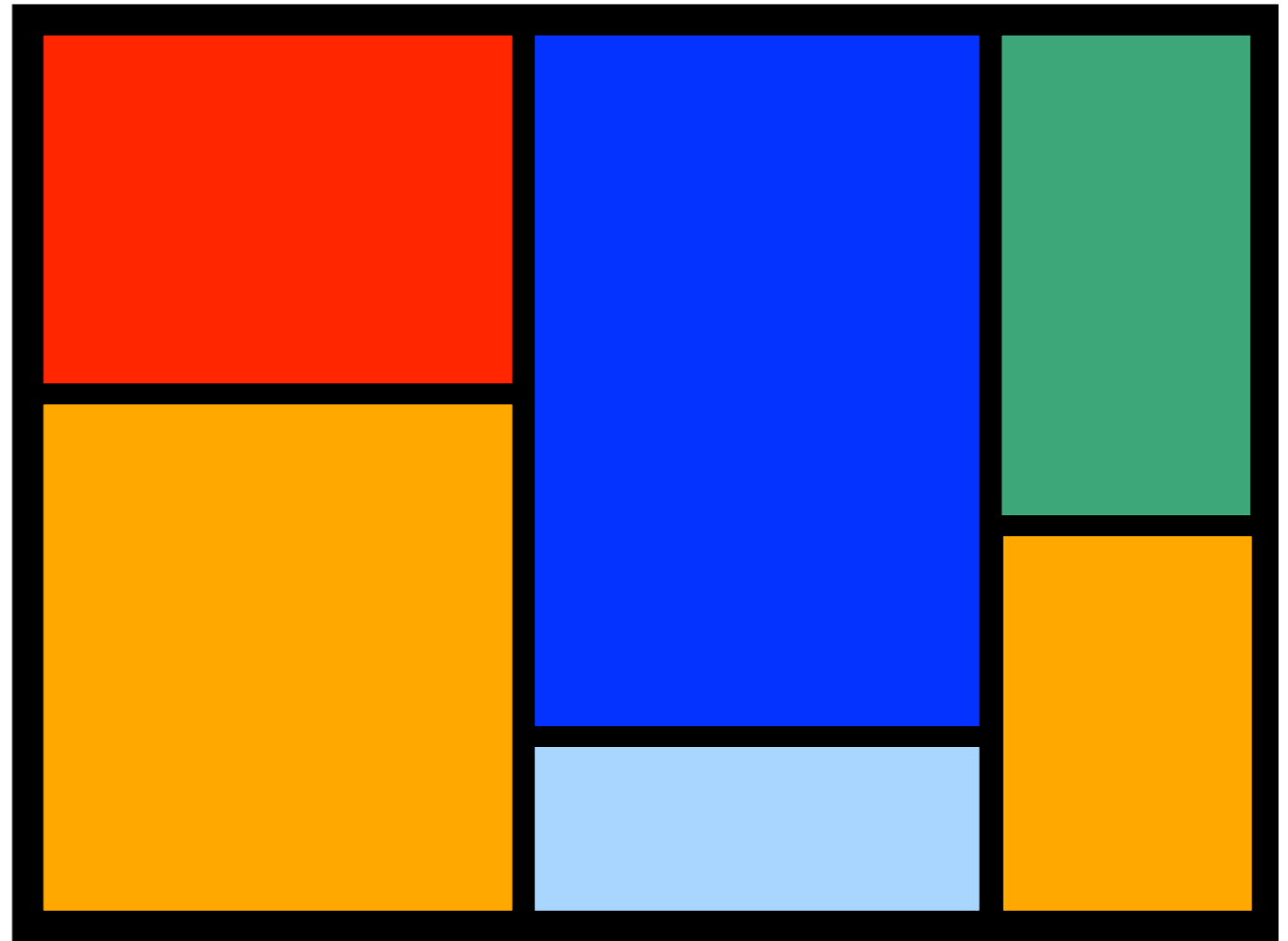
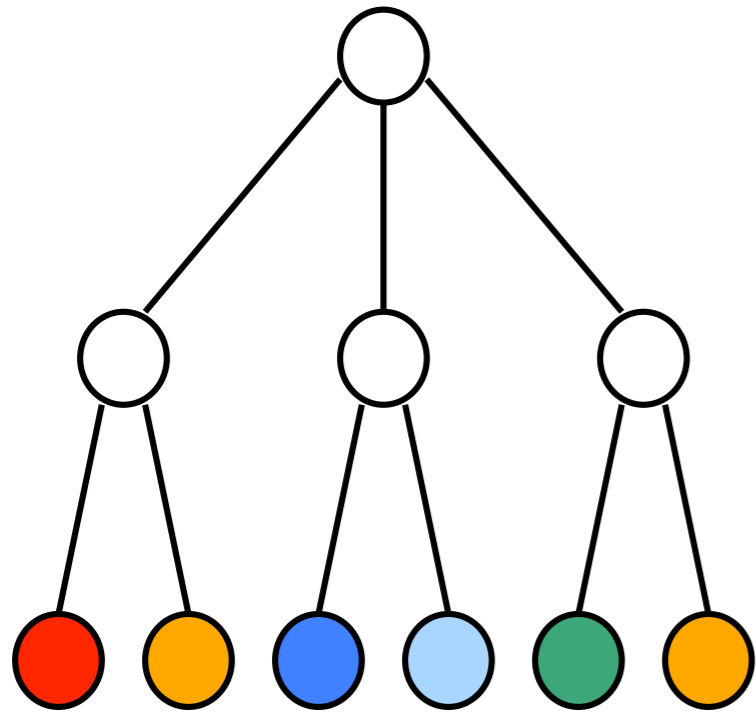
Traditional Approaches to
Software Visualization



Unified Modelling Language (UML)



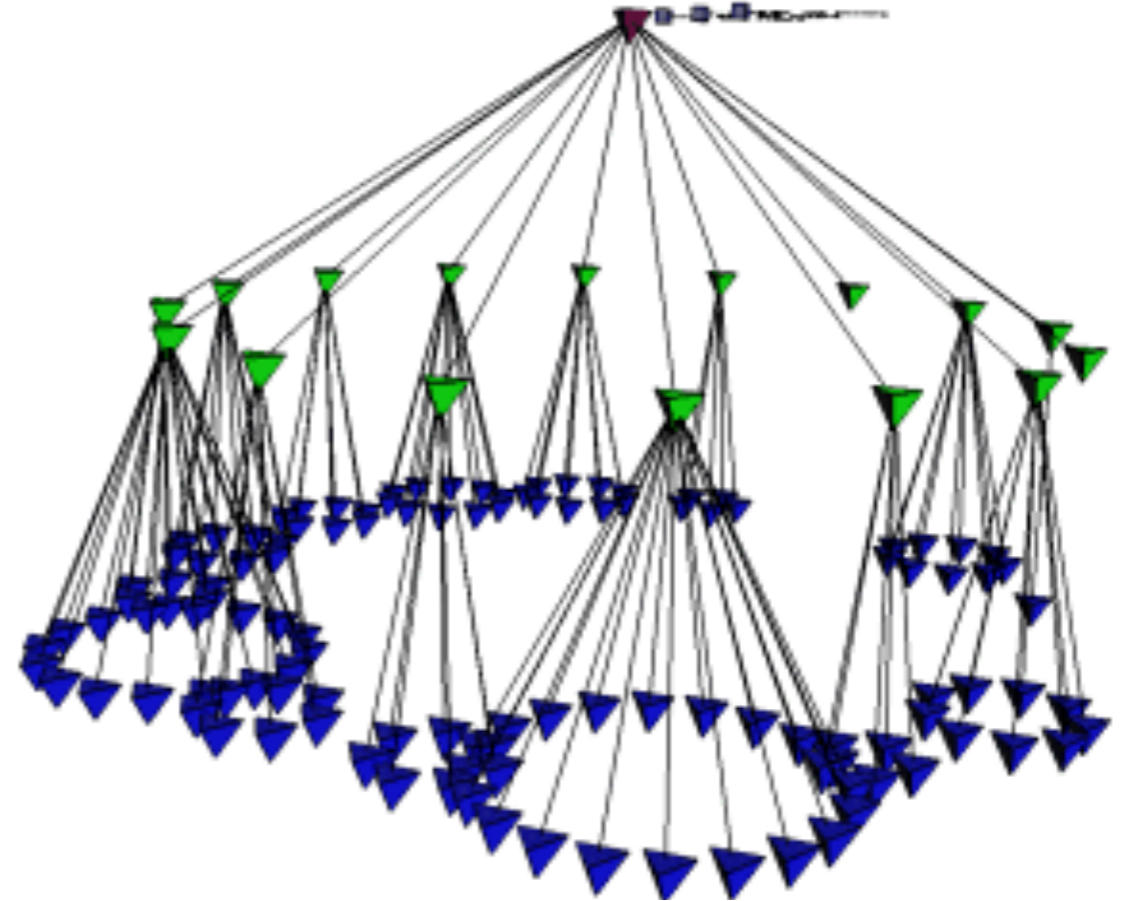
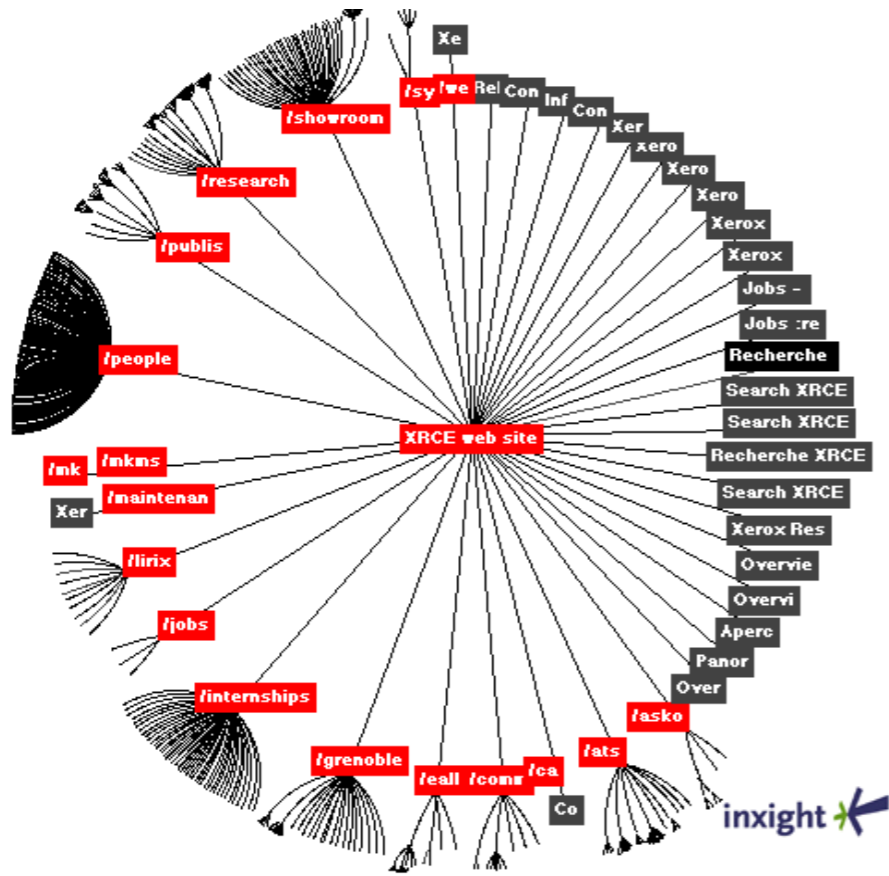
Unified Modelling Language (UML)



Treemaps

The screenshot shows a classic Macintosh desktop with a red background. A large window in the center displays a directory listing with the following columns: Name, Size, % Total, Type, Creator, Creation Date, and Modification Date. The listing includes various files and folders such as 'Bella Rosa', 'The Ha...', 'Backgro...', 'Sony VTR', 'Navigator', 'MECH', 'Noh_Tale', 'Shared.DIR', 'Chair', 'Rollover', 'Use', 'wallcov', 'Ink_FX', 'shared.dir', 'Weeping', 'ATM™ GX', 'QuickDraw™', and 'Skia'. The desktop also features several other windows, including 'Director 4.0', 'Director 4.0 Help', 'HyperCard Player', 'Audio Help', 'PowerTalk', 'Mail', 'User Guide', 'Installer', and 'Apple DocViewer'. The desktop is cluttered with many small icons and windows, representing a typical Macintosh environment from the early 1990s.

Name	Size	% Total	Type	Creator	Creation Date	Modification Date
Unknown	Text	Graphics	Archives/Stacks	Programming	Applications	System



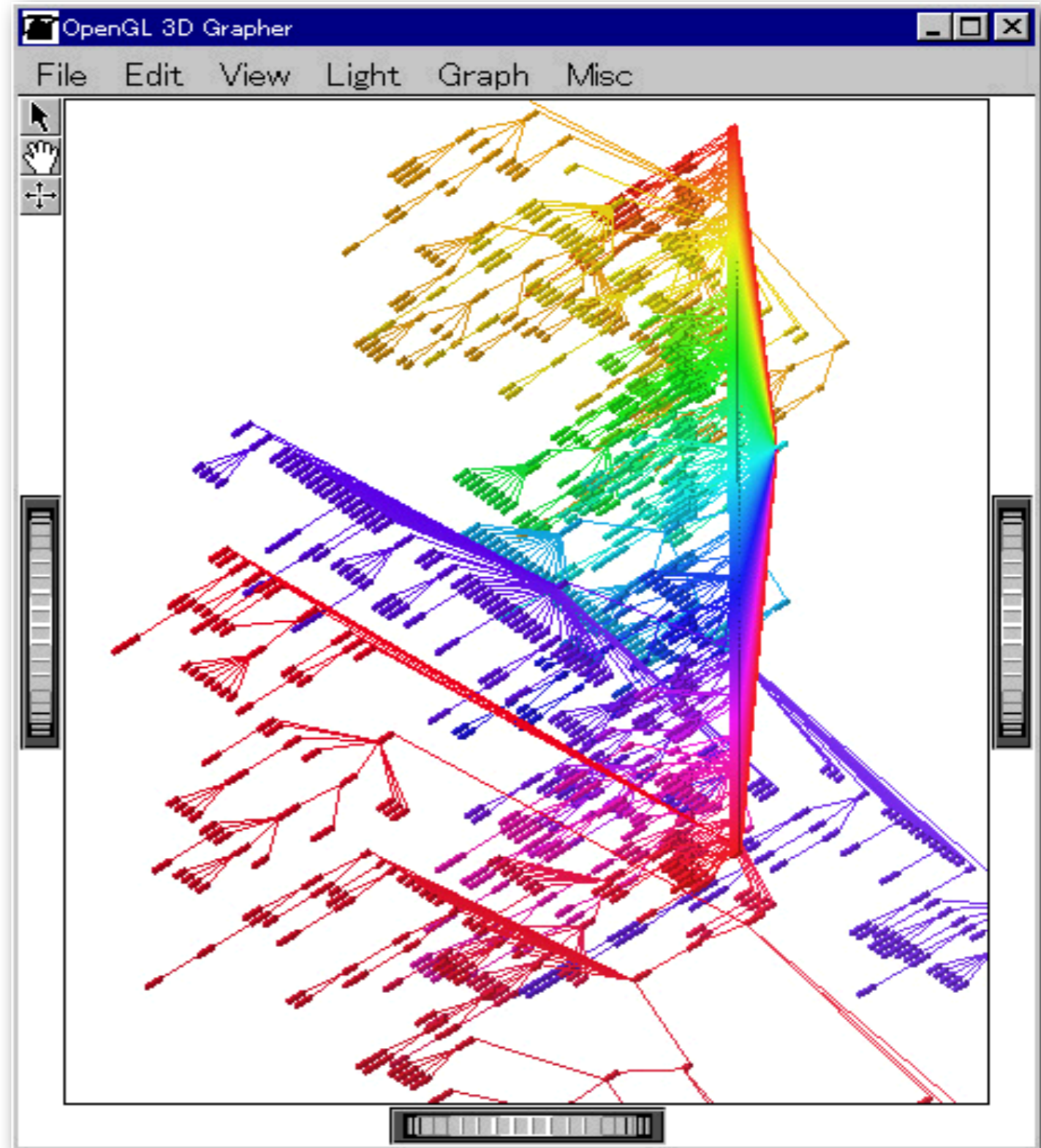
Euclidian Cones and Hyperbolic Trees

Jun/OpenGL

Graph on the right displays the class hierarchy of Smalltalk

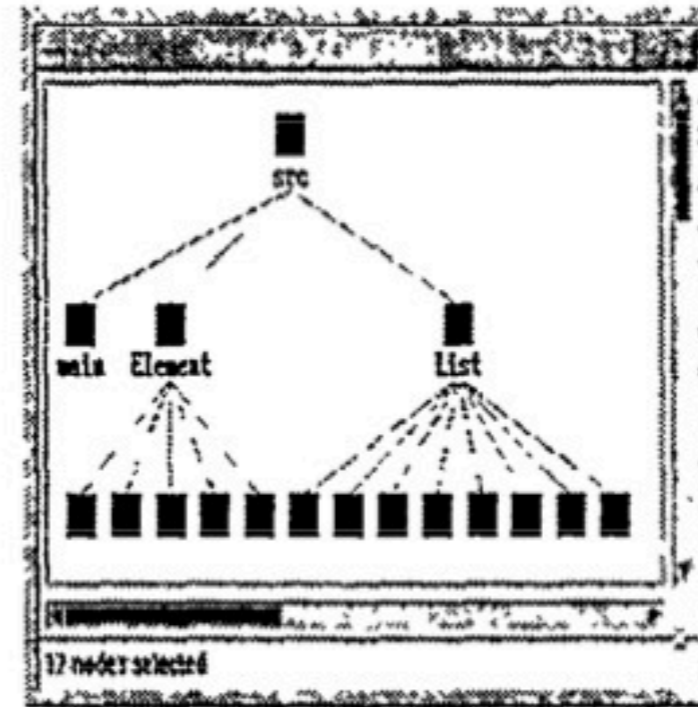
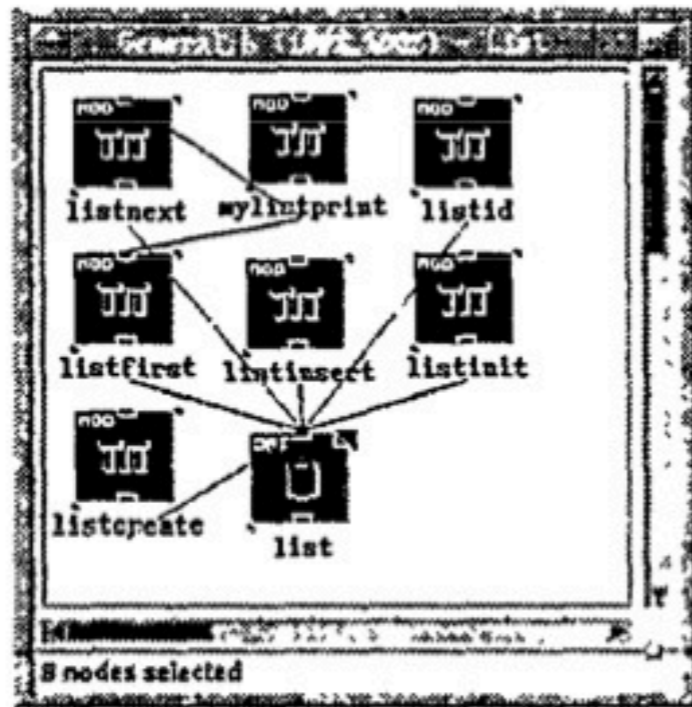
Problems:

- Visual **overload**
- **Colors** are meaningless
- **Navigation** is difficult





Rigi



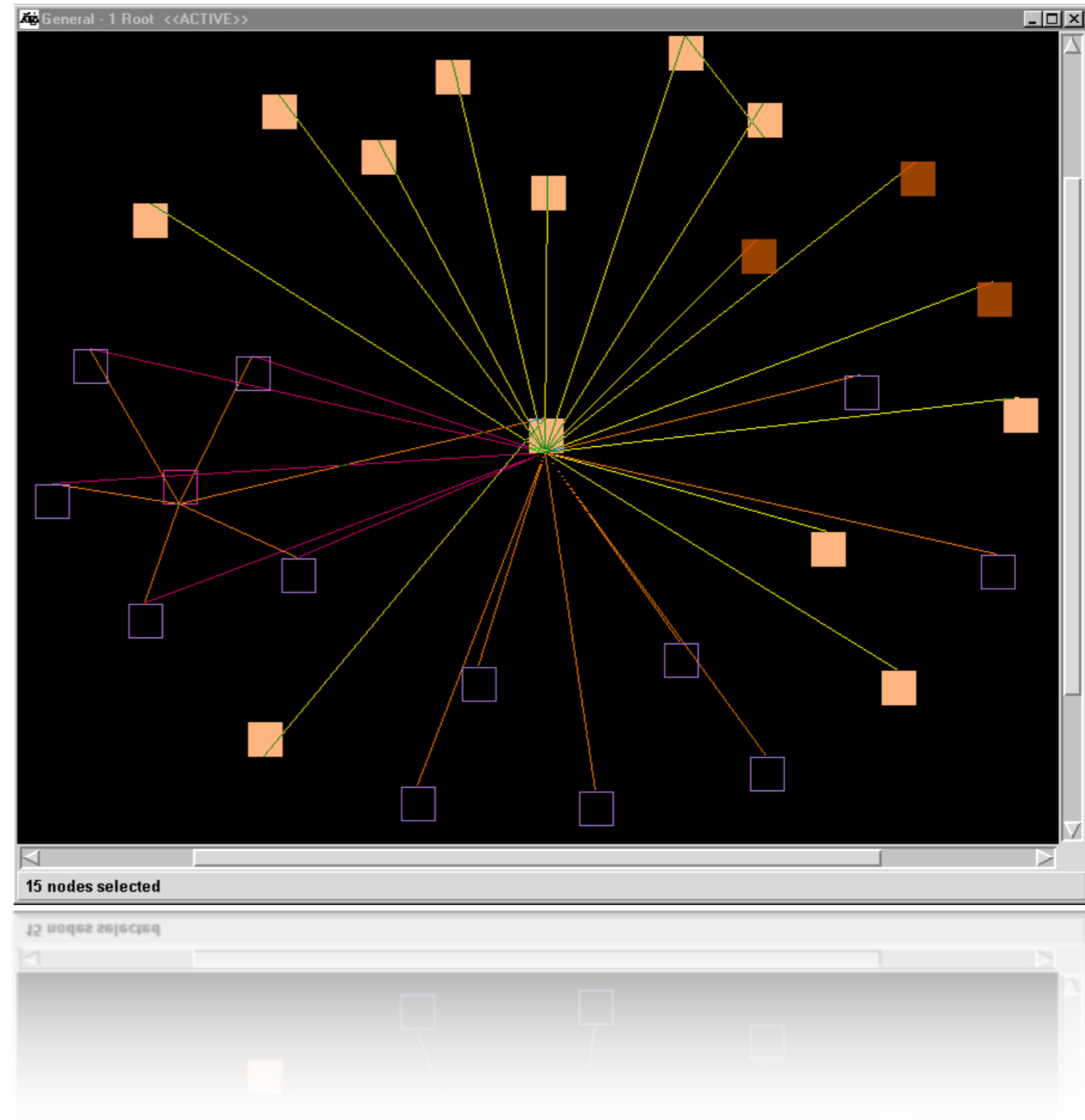
That's how it is supposed to look like!

Rigi

Entity-Relationship Visualization

Allows grouping and aggregation of entities

Problem: Still lacks of code semantics



Shrimp/Creole

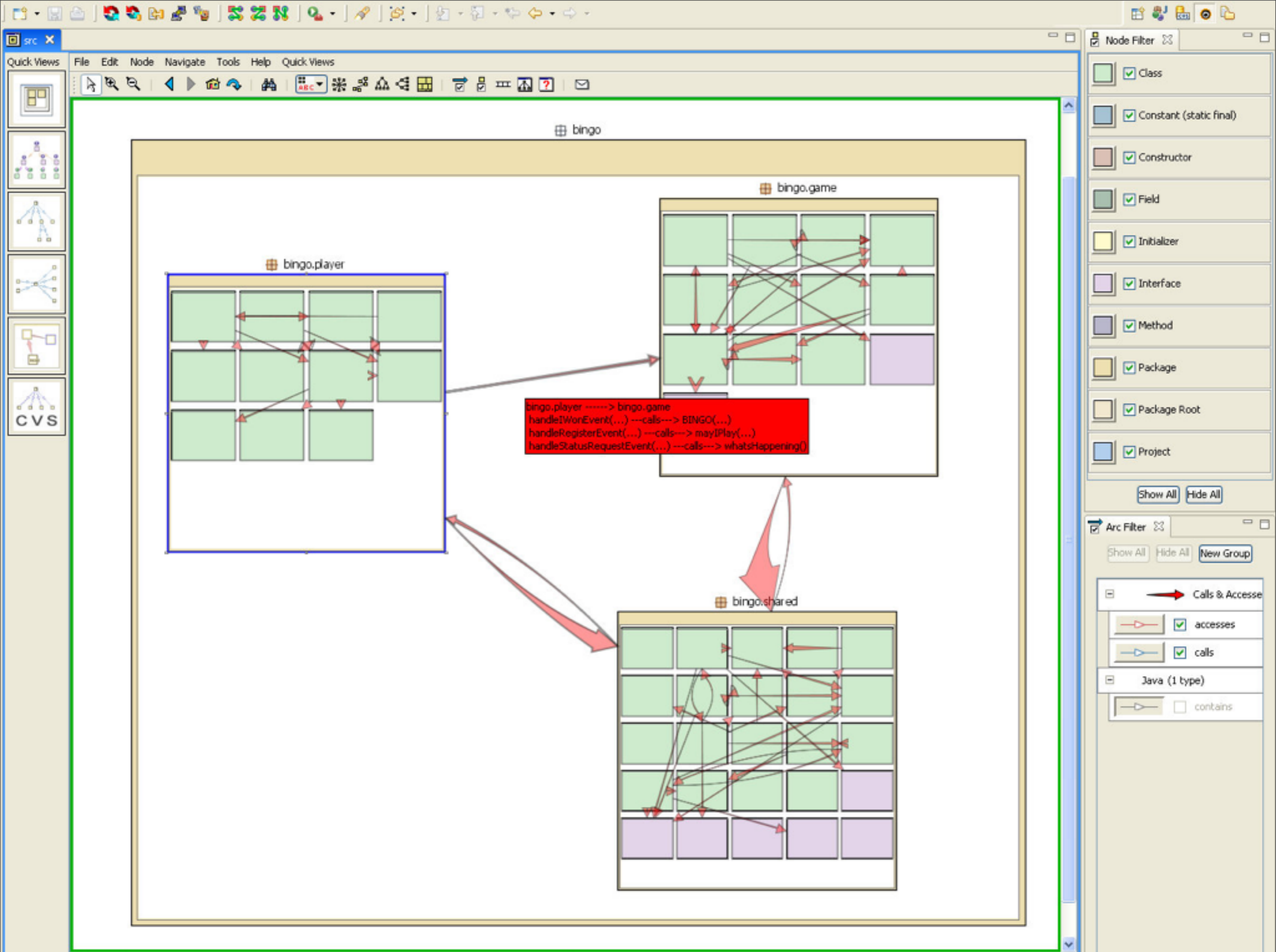
<http://www.thechiselgroup.org/creole>

SHriMP (**S**imple **H**ierarchical **M**ulti-**P**erspective) is a domain-independent visualization technique.

Creole is a top-down approach to visualize Java source code within the Eclipse IDE.

Problem: Graph is cluttered with irrelevant details.





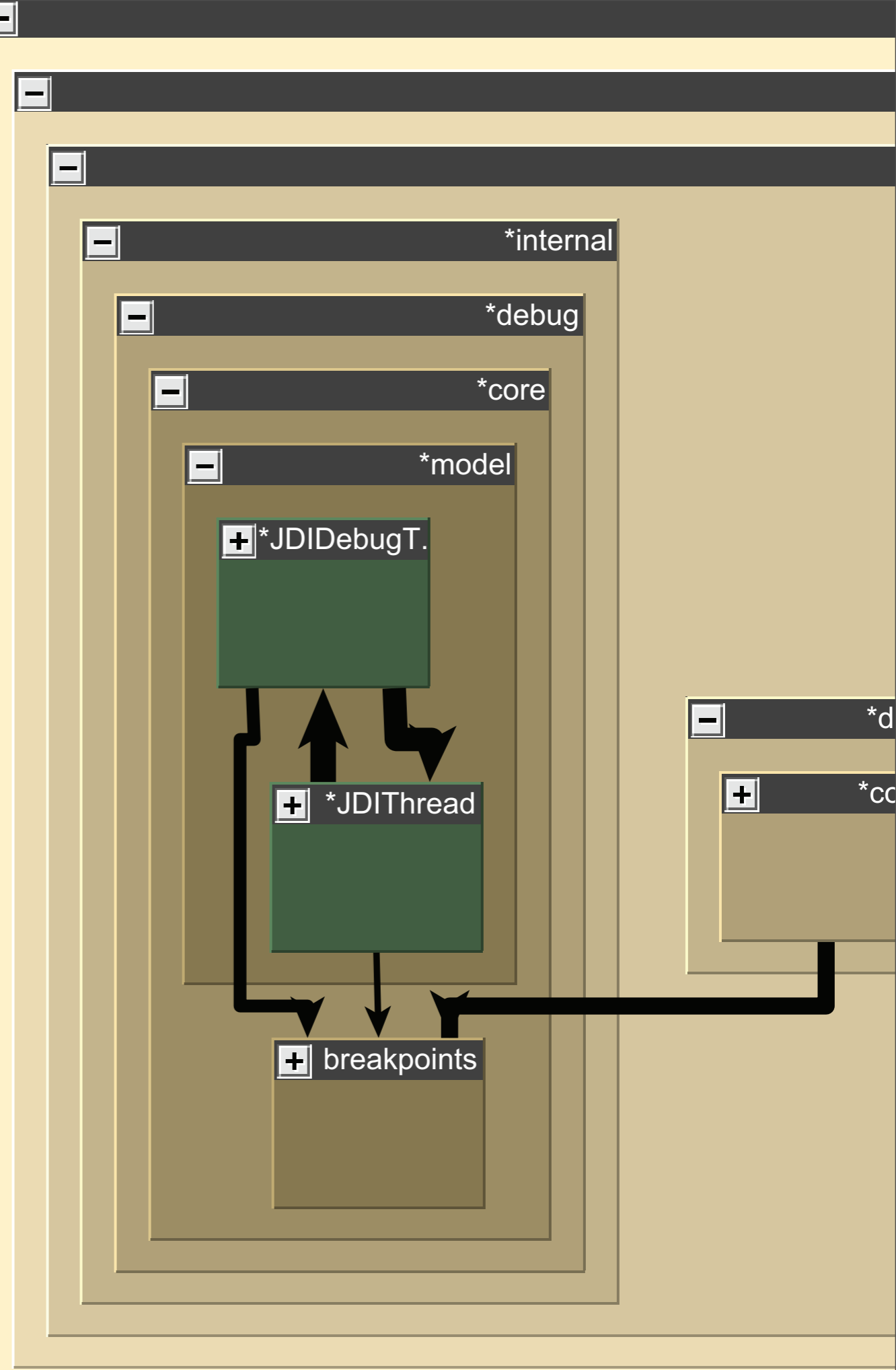
DA4Java

<http://serg.aau.at/bin/view/MartinPinzger/DA4Java>

DA4Java uses nested graphs to visualize source code in the Eclipse IDE.

Allows to incrementally add and filter source code information.

Combines top-down with bottom-up analysis for insights on overall structure and gory details



Dependency Analyzer - Dependency Graph - Eclipse SDK

Package Explorer

- org.eclipse.compare
- org.eclipse.jdt.debug
- org.eclipse.osgi R3_2 [dev.eclipse.org]
- org.evolizer.core.hibernate [core/trunk/org.evolizer.core.]
- org.evolizer.core.logging [core/trunk/org.evolizer.core.]
- org.evolizer.da4java [da4java/trunk/org.evolizer.da4java]
- JRE System Library [J2SE-1.5]
- Plug-in Dependencies
- src
 - org.evolizer.da4java
 - org.evolizer.da4java.commands
 - org.evolizer.da4java.commands.additions
 - org.evolizer.da4java.commands.filters
 - org.evolizer.da4java.graph.data
 - org.evolizer.da4java.graph.panel
 - org.evolizer.da4java.graph.panel.edgerendere
 - org.evolizer.da4java.graph.panel.images
 - org.evolizer.da4java.graph.panel.noderendere
 - org.evolizer.da4java.graph.panel.toolbaraction
 - org.evolizer.da4java.graph.utils
 - org.evolizer.da4java.plugin
 - AddSelectedEntitiesAction.java 1463 1/30
 - AddSelectedEntitiesJob.java 1675 3/10/0
 - LoadAndShowGraphJob.java 1640 2/22/0
 - LoadAndShowGraphJob 1640 2/22/0
 - fMonitor

SnapshotAnalyzer.java | Activator.java | DependencyGraph.java | org.evolizer.da4java

Layout: Incremental Hierarchic

Polymetric View Controller

Polymetric View Control

Profiles

Available Profiles: [Dropdown]

Save Delete

Metric Dimension Selectors

Node Height: Uniform [Dropdown]

Node Width: Uniform [Dropdown]

Node Color: Uniform [Dropdown]

Refresh Graph

Entity Visibility

Entity Visibility Control

Entity Type

- Package
- Class
- Method
- Attribute




Association Visibility

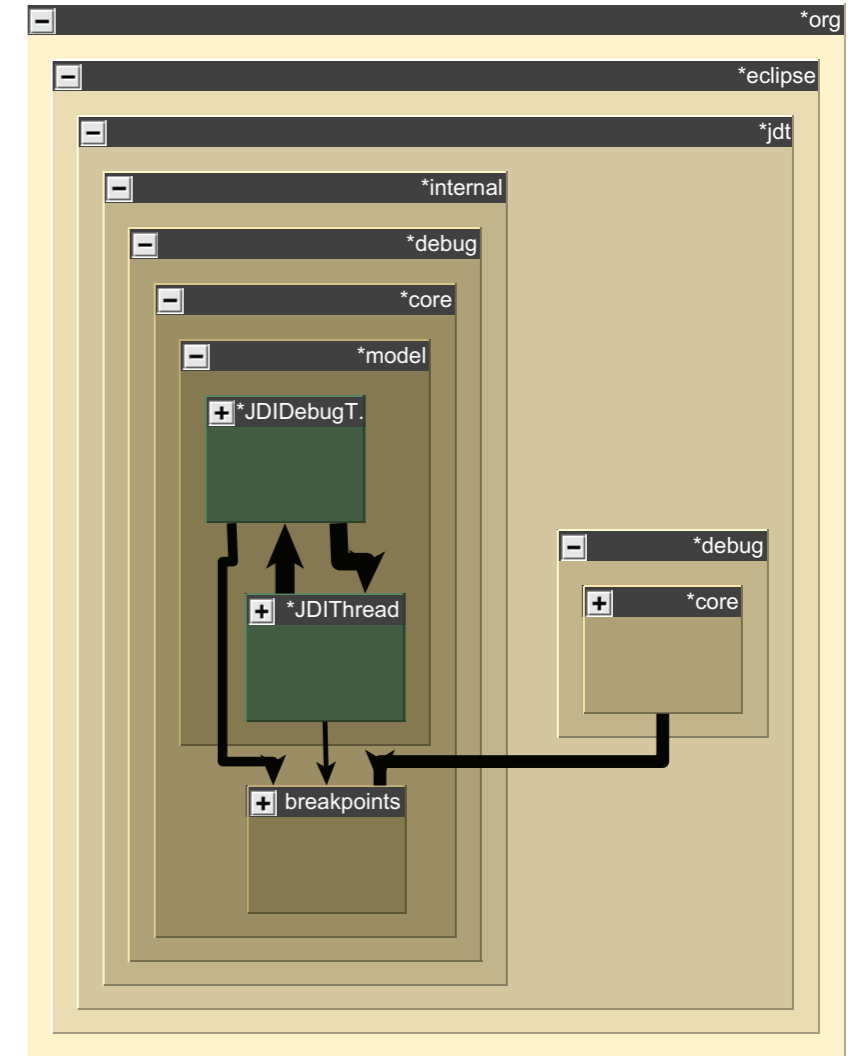
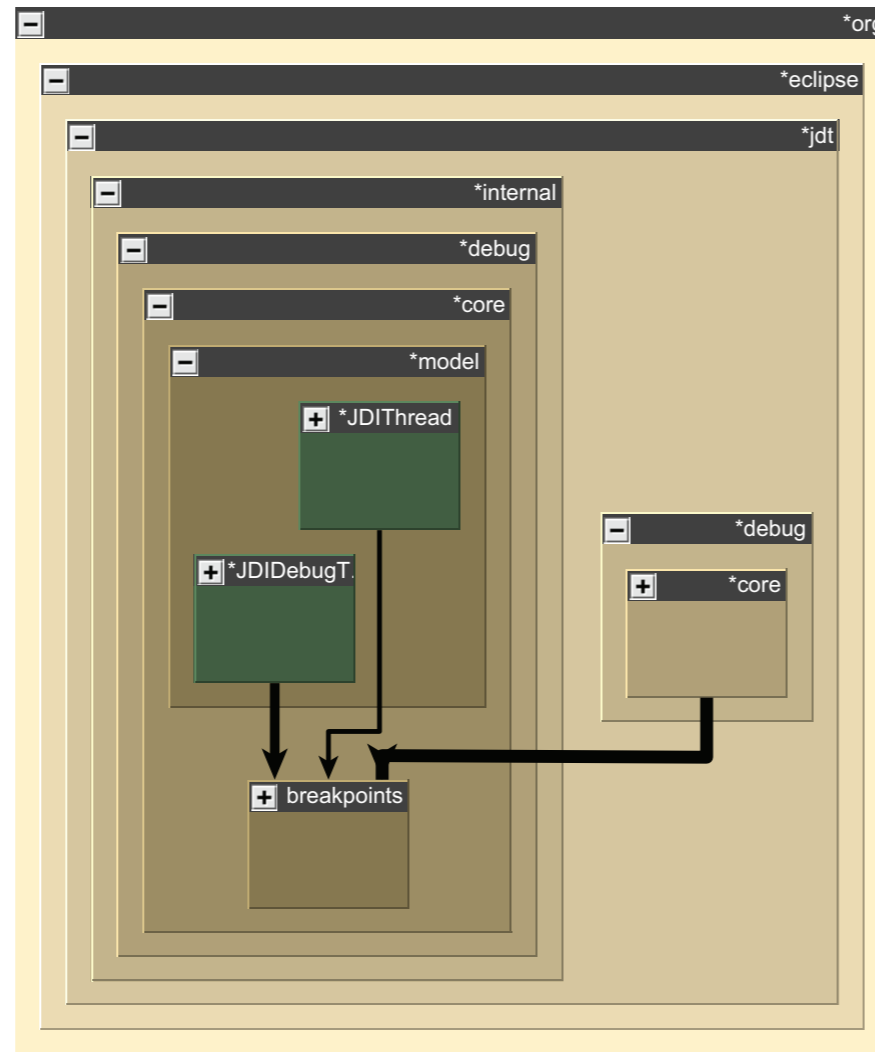
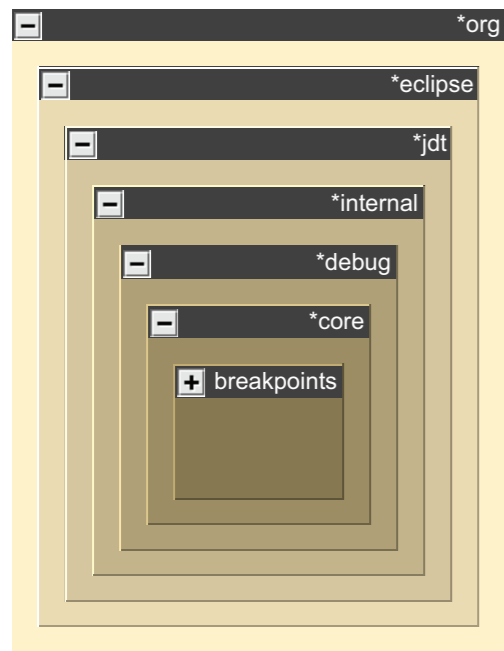
Association Visibility Control

Association Visibility

- Accesses
- Invocations
- Inherits
- Subtypes
- Cast tos
- Check Instance ofs

DA4Java in action within the Eclipse IDE

-  Package
-  Java class
-  Method call



- Step 1) Add package 'breakpoints'
- Step 2) Add callers of package 'breakpoints'
- Step 3) Added method calls between classes 'JDIDebugTarget' and 'JDIThread'

Polymetric Views

<http://www.inf.usi.ch/faculty/lanza/codecrawler.html>

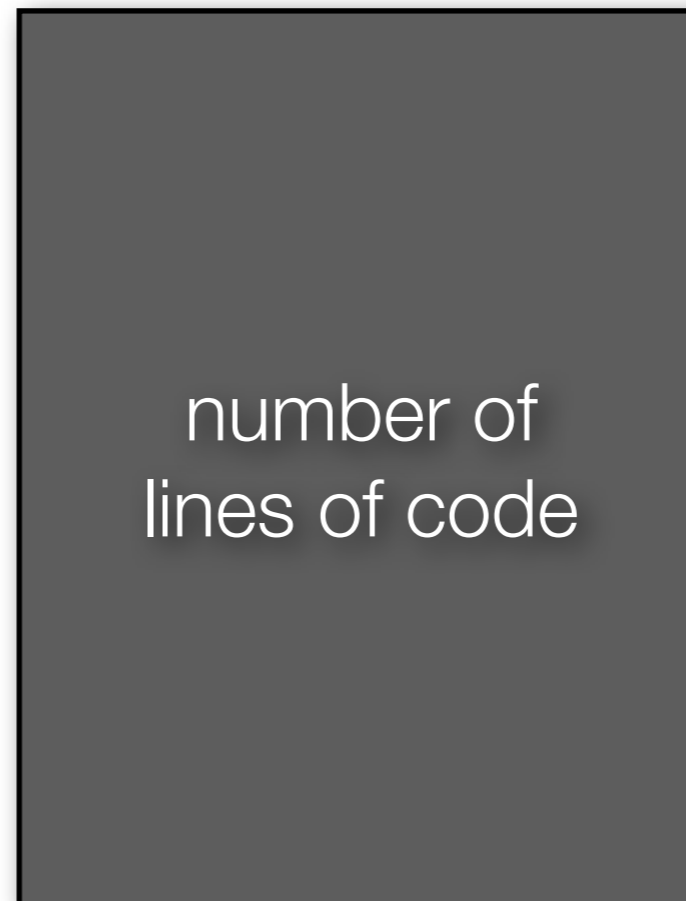
Visualizations of a graph enriched with (multiple) **metrics**.

System Complexity is a polymetric view that shows the classes of the system, organized in **inheritance hierarchies**.

Each class is represented by a node.

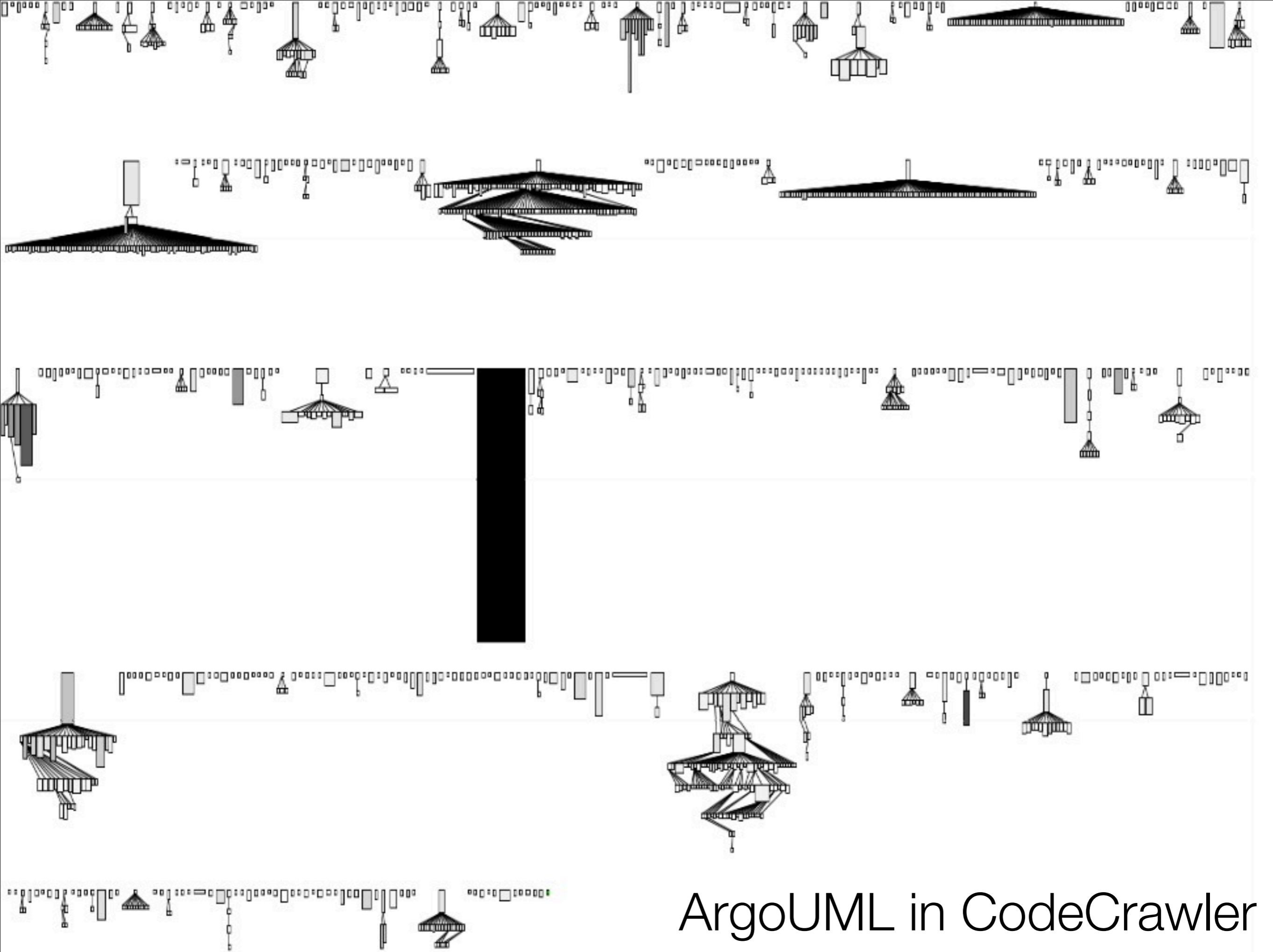
Edges denote inheritance relationships.

number of attributes

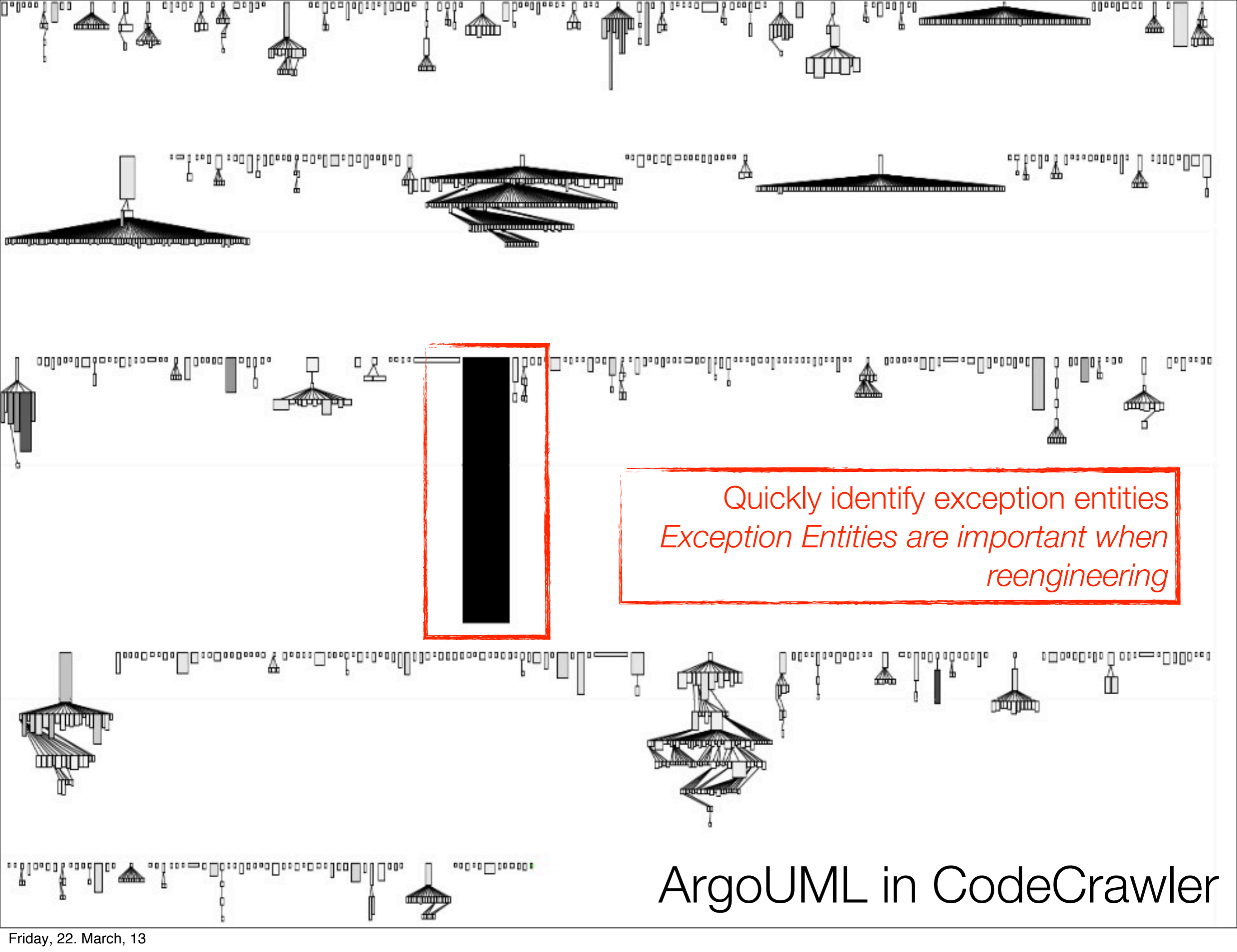


number of lines of code

number of methods

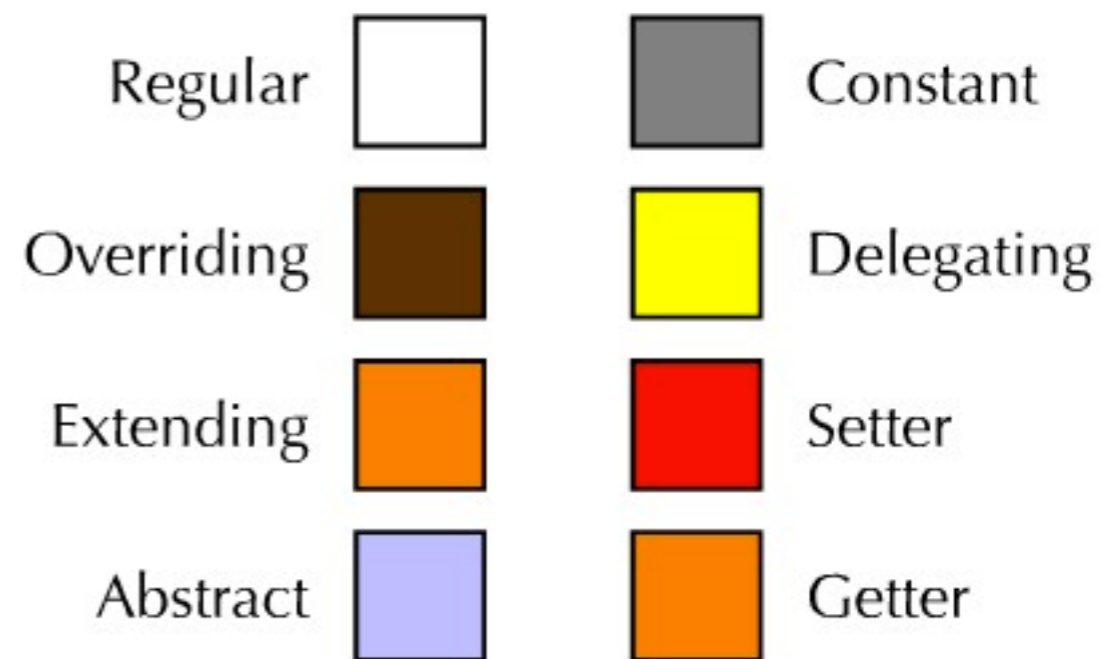
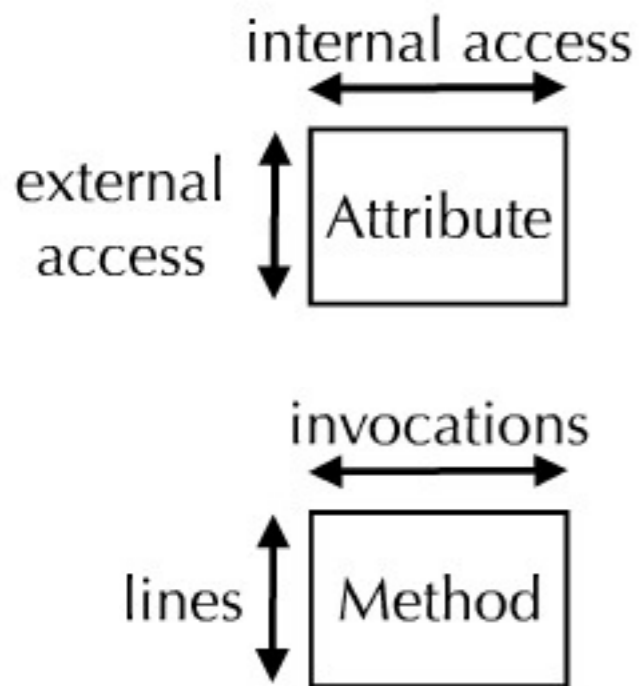
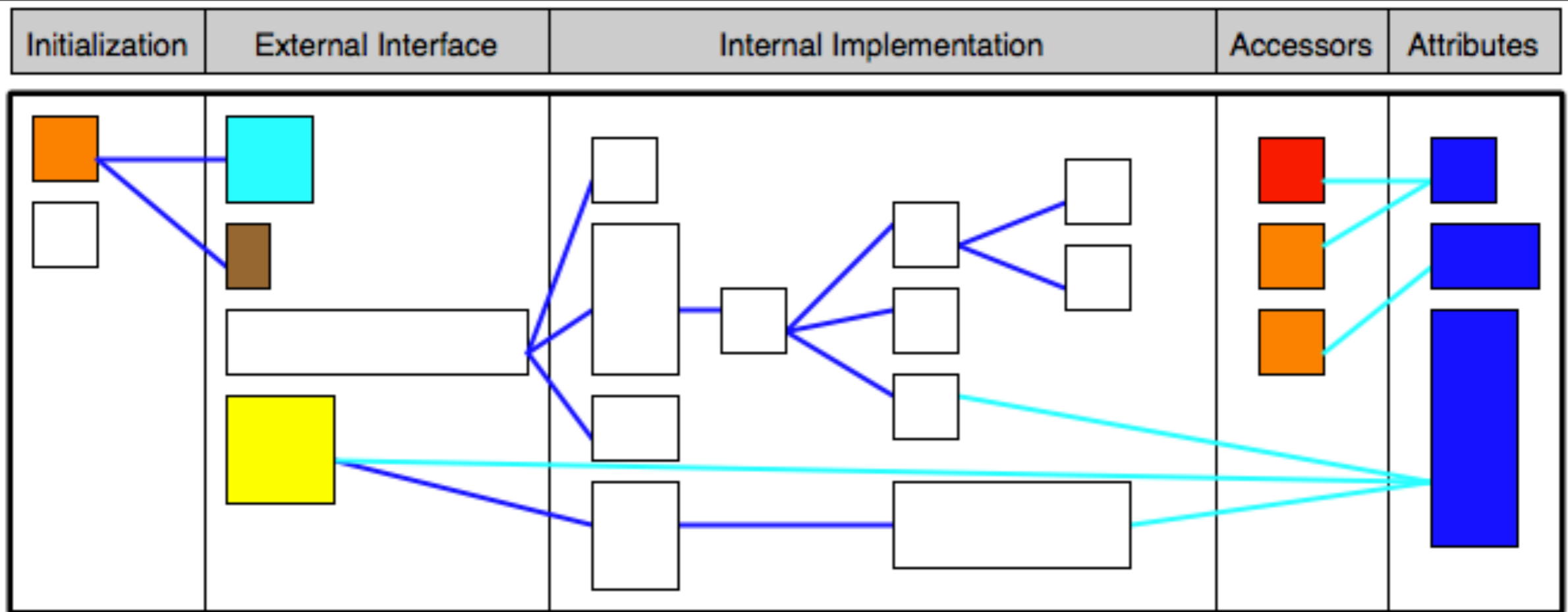


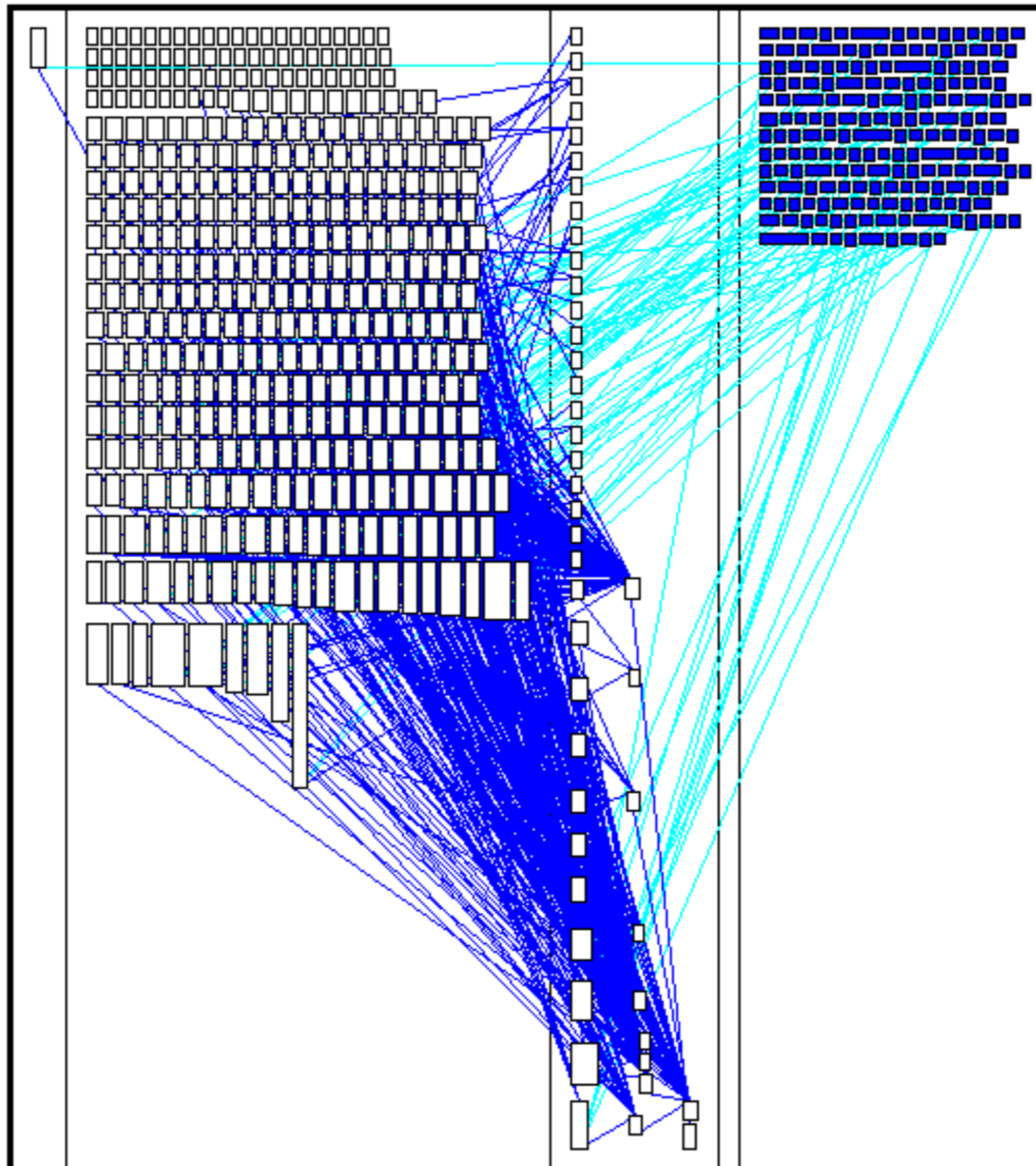
ArgoUML in CodeCrawler



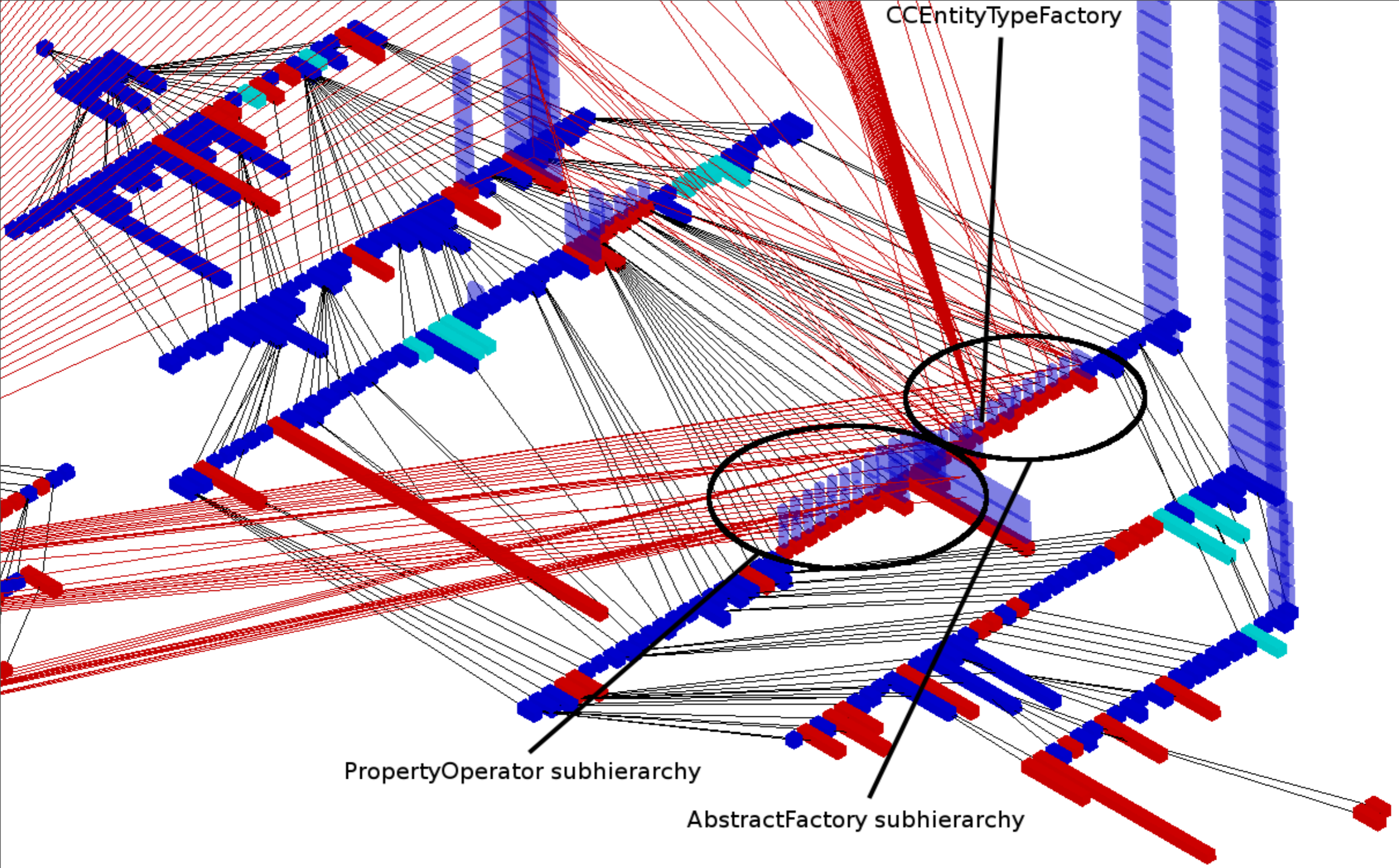
Quickly identify exception entities
Exception Entities are important when reengineering

ArgoUML in CodeCrawler





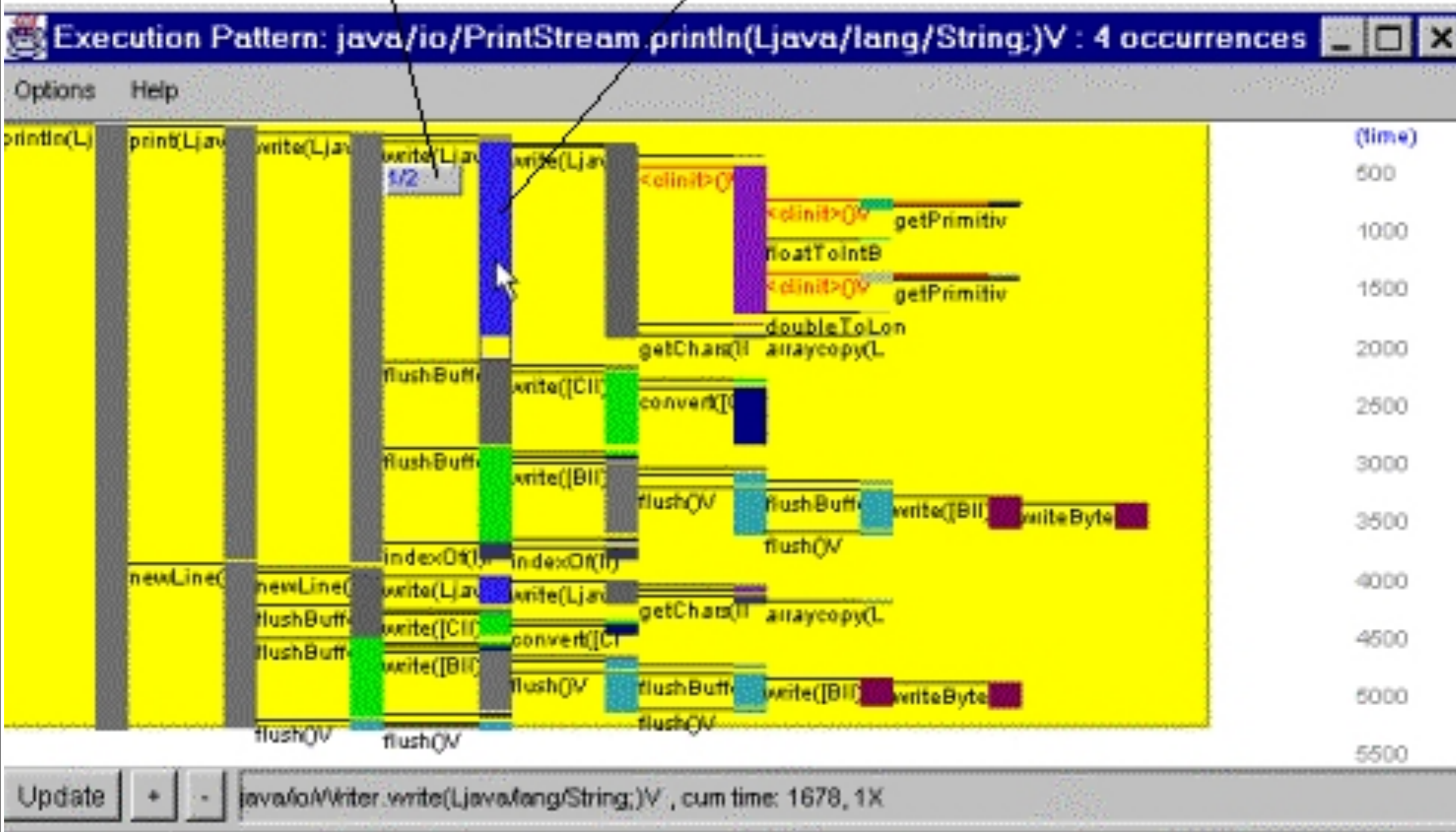
An old friend pays us a visit: ArgoUML's ModelFacade
453 Methods - 114 Attributes - 3500 LOC - Coupled to hundreds of classes



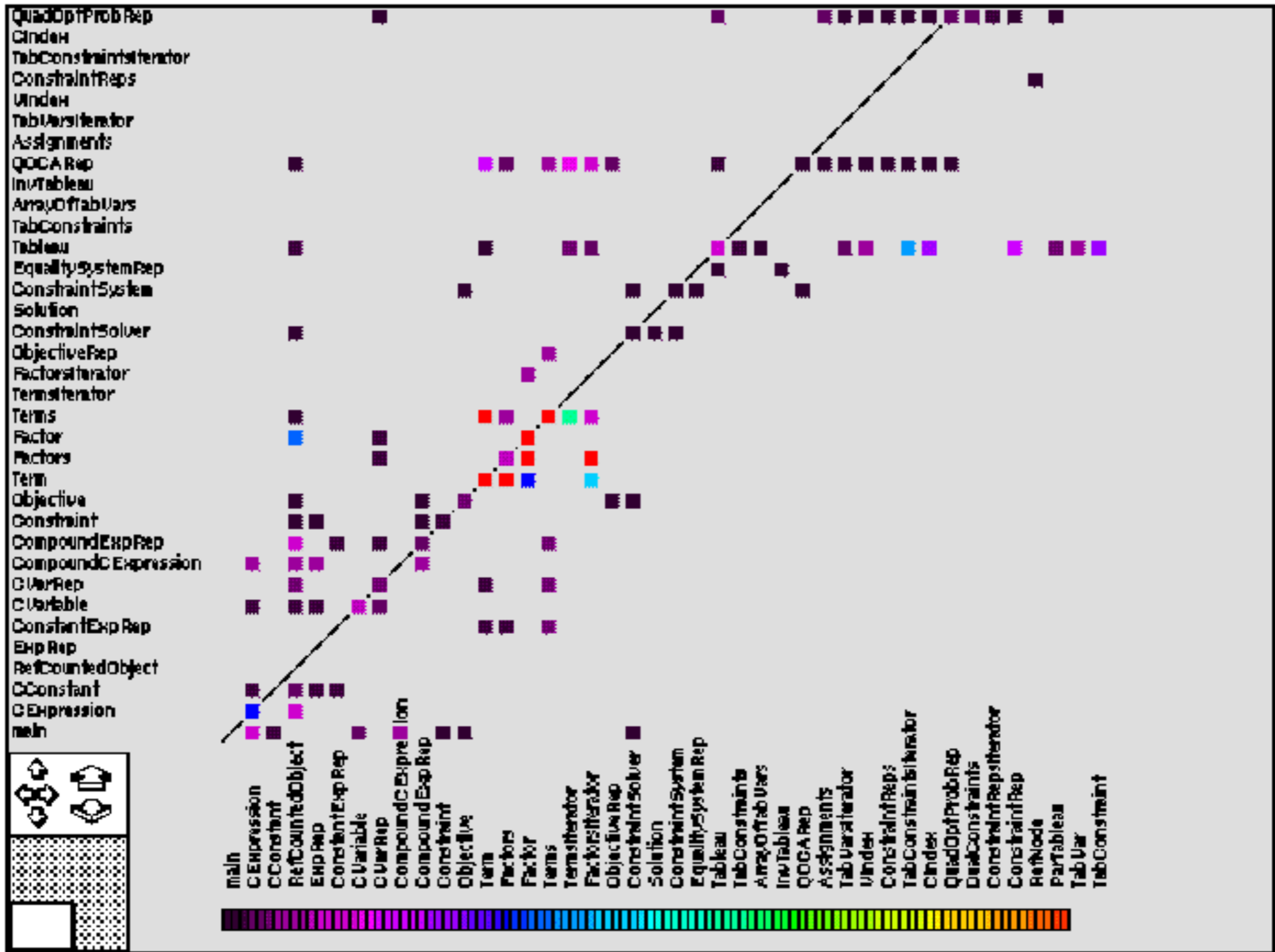
CodeCrawler, Runtime Visualization, Instance Collaboration View

variant button

bright blue stripe



JInsight - Visualization of an execution stack trace



Inter-class call matrix

Evolution Radar

Reference module is placed in center.

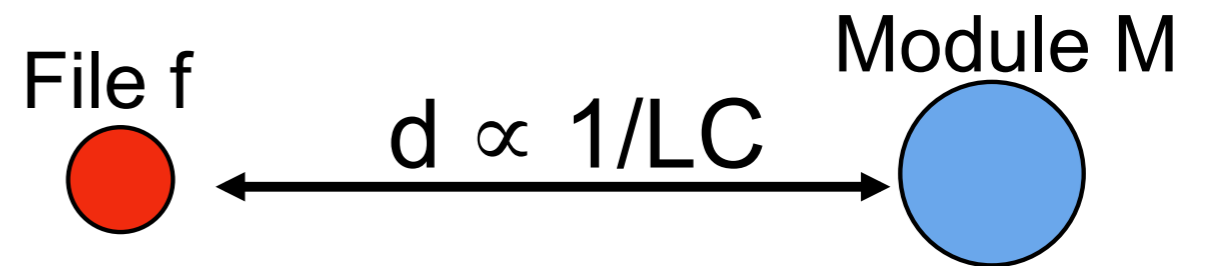
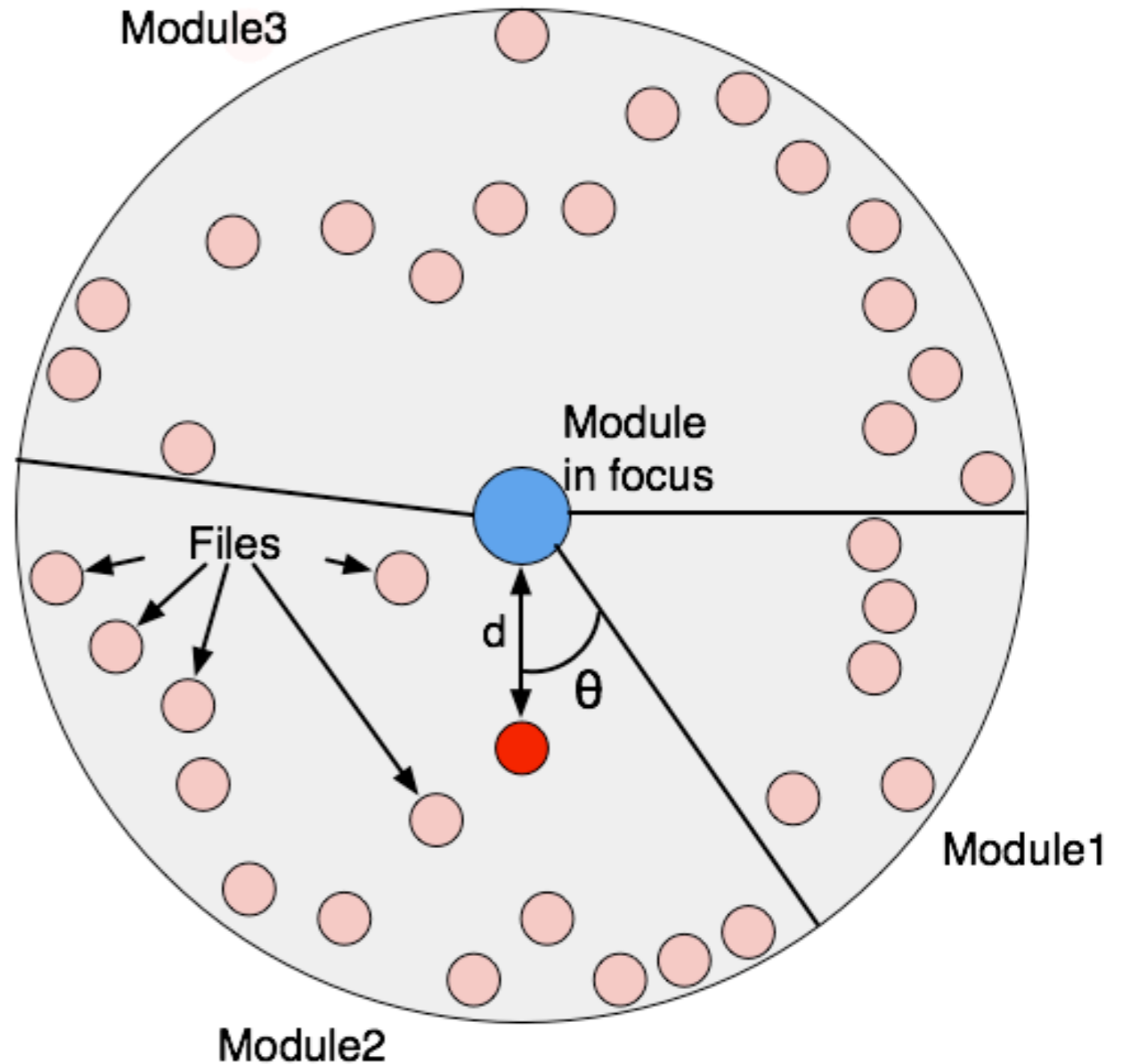
All other modules are shown as sectors.

For each module, all its files are rendered as colored circles and positioned using polar coordinates:

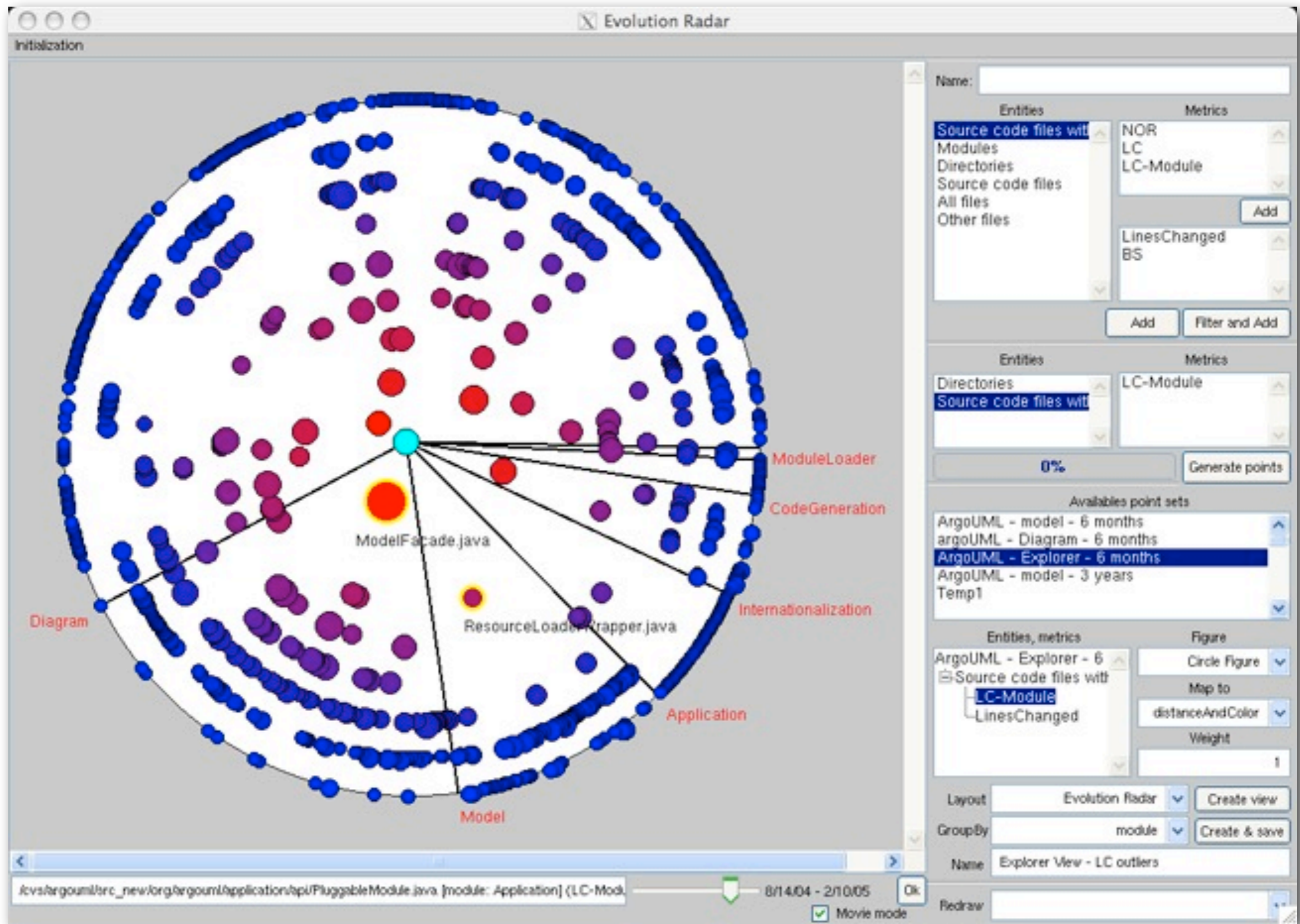
d : inverse proportional to LC

Θ : alphabetical sorting and uniform distribution

LC is the number of shared commits.



$$LC(M, f) = \max_{f_i \in M} LC(f_i, f)$$



ArgoUML in Evolution Radar - <http://www.inf.usi.ch/phd/dambros/tools/evoradar.php>

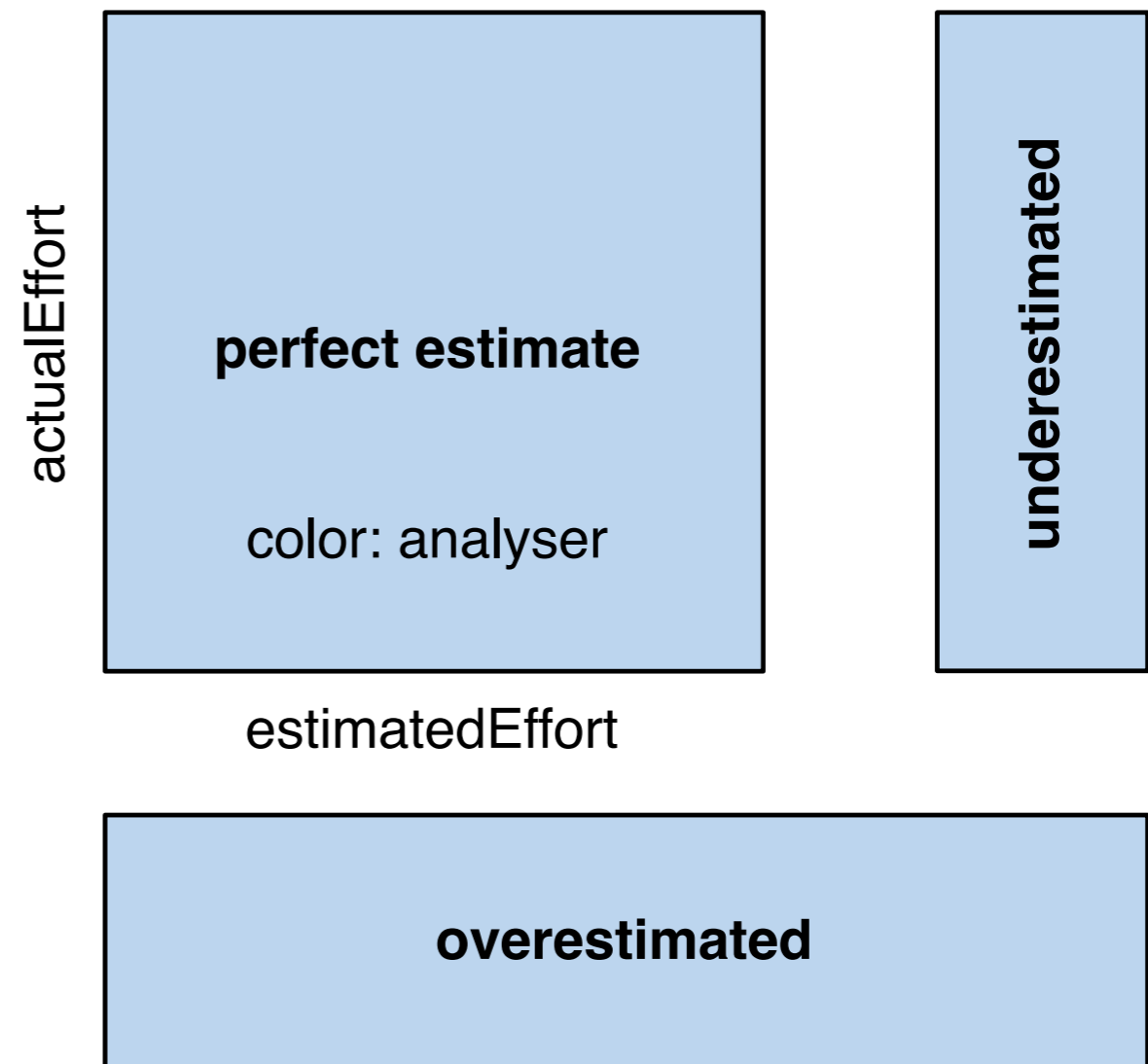
Smart Views

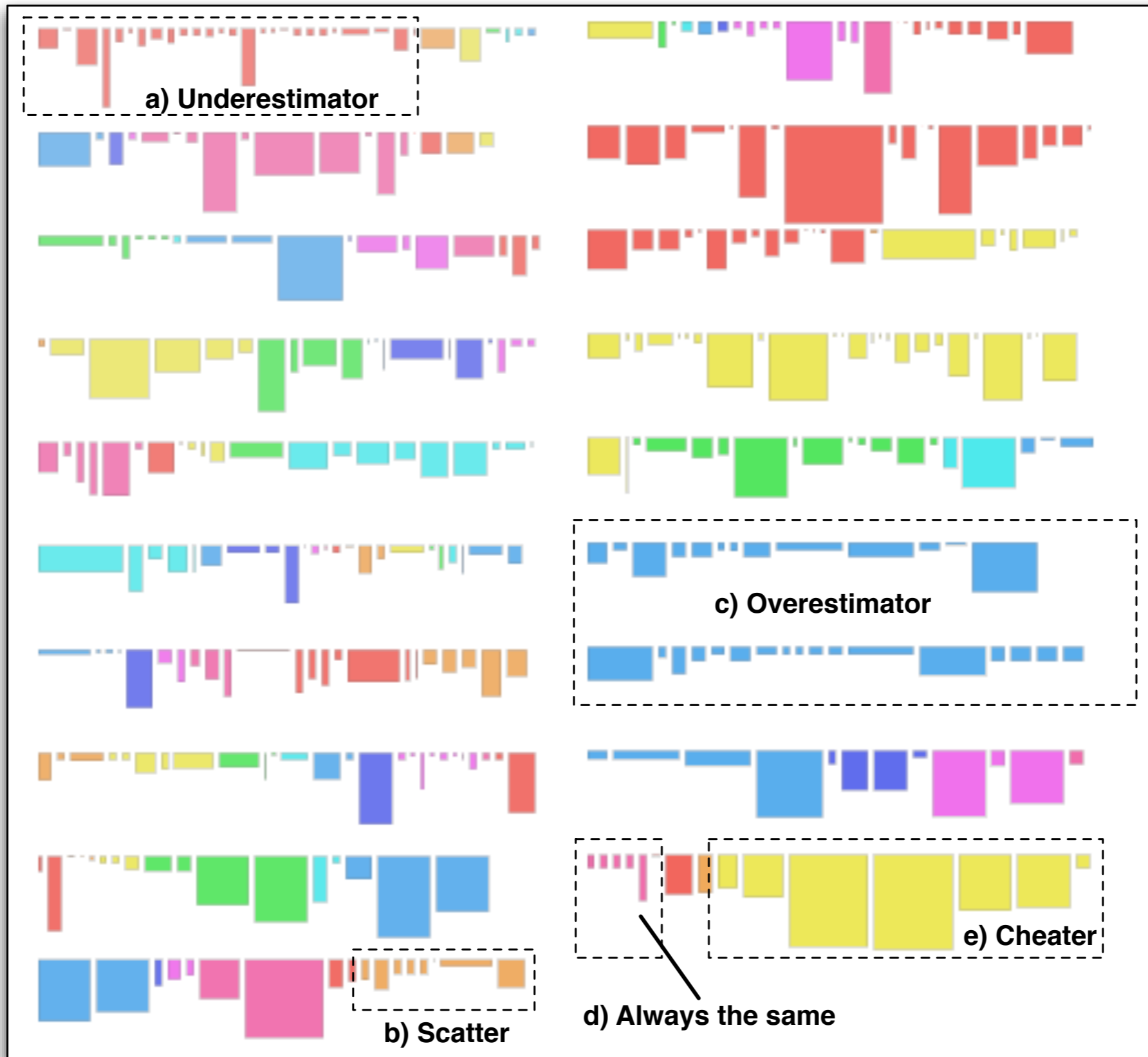
<http://seal.ifi.uzh.ch/>

Polymetric View on effort estimation data.

Visualizes effort estimation patterns.

Can be used to measure and improve the efficiency and effectiveness of the development process.





Smart Views: Effort View on PRs grouped and colored according to the analyzer



About Metaphors

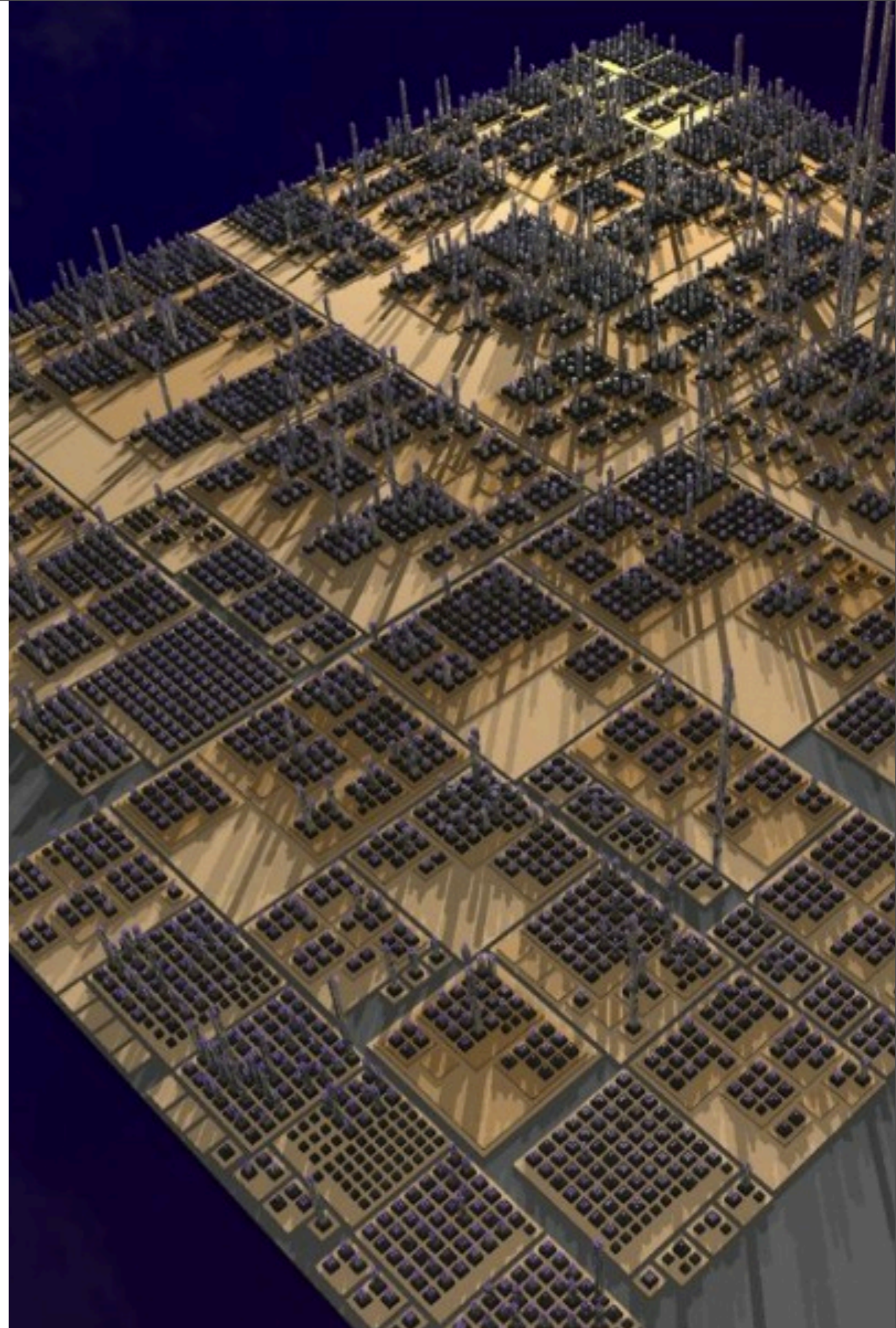
Software Visualization++

CodeCity

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

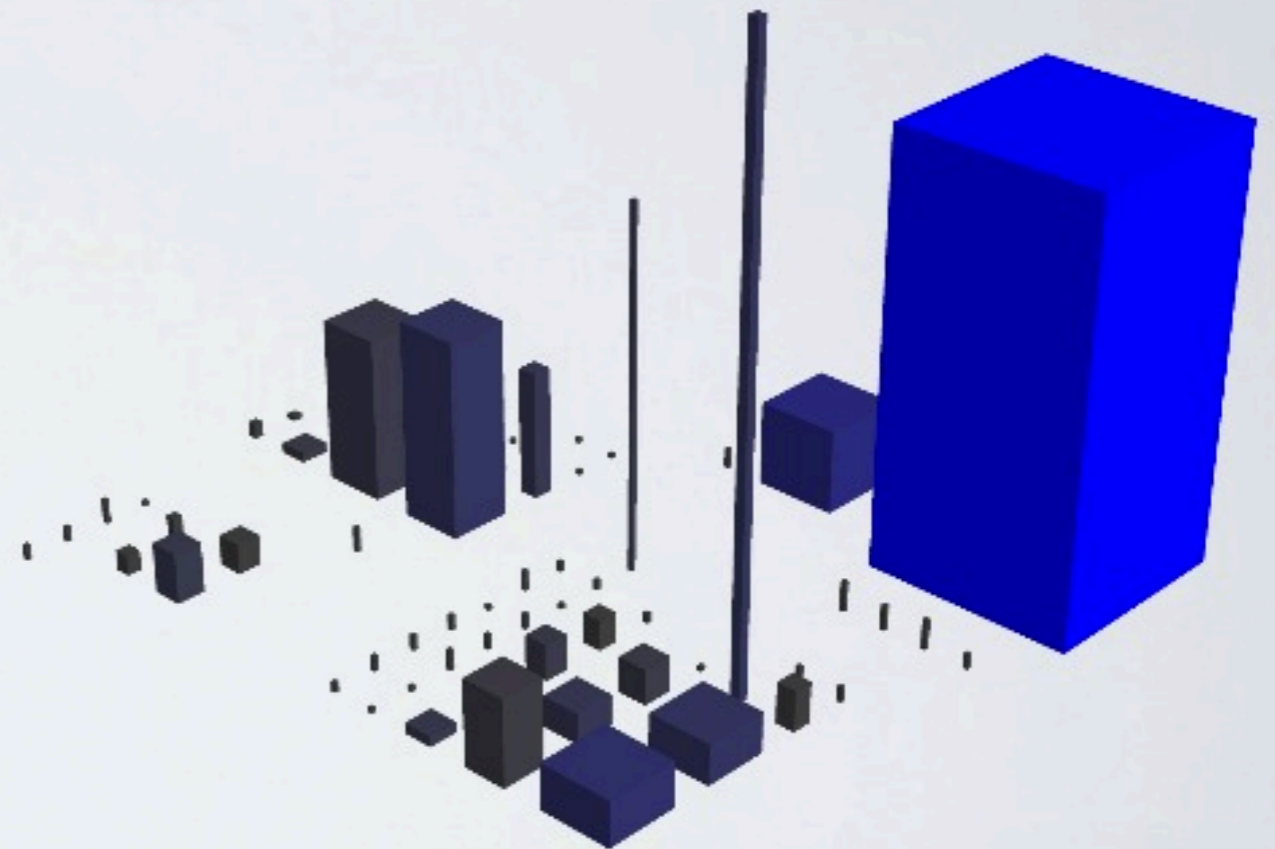
Software systems are visualized as interactive, navigable 3D cities

The visible properties of the city artifacts depict a set of chosen software metrics

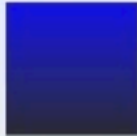


Our city metaphor

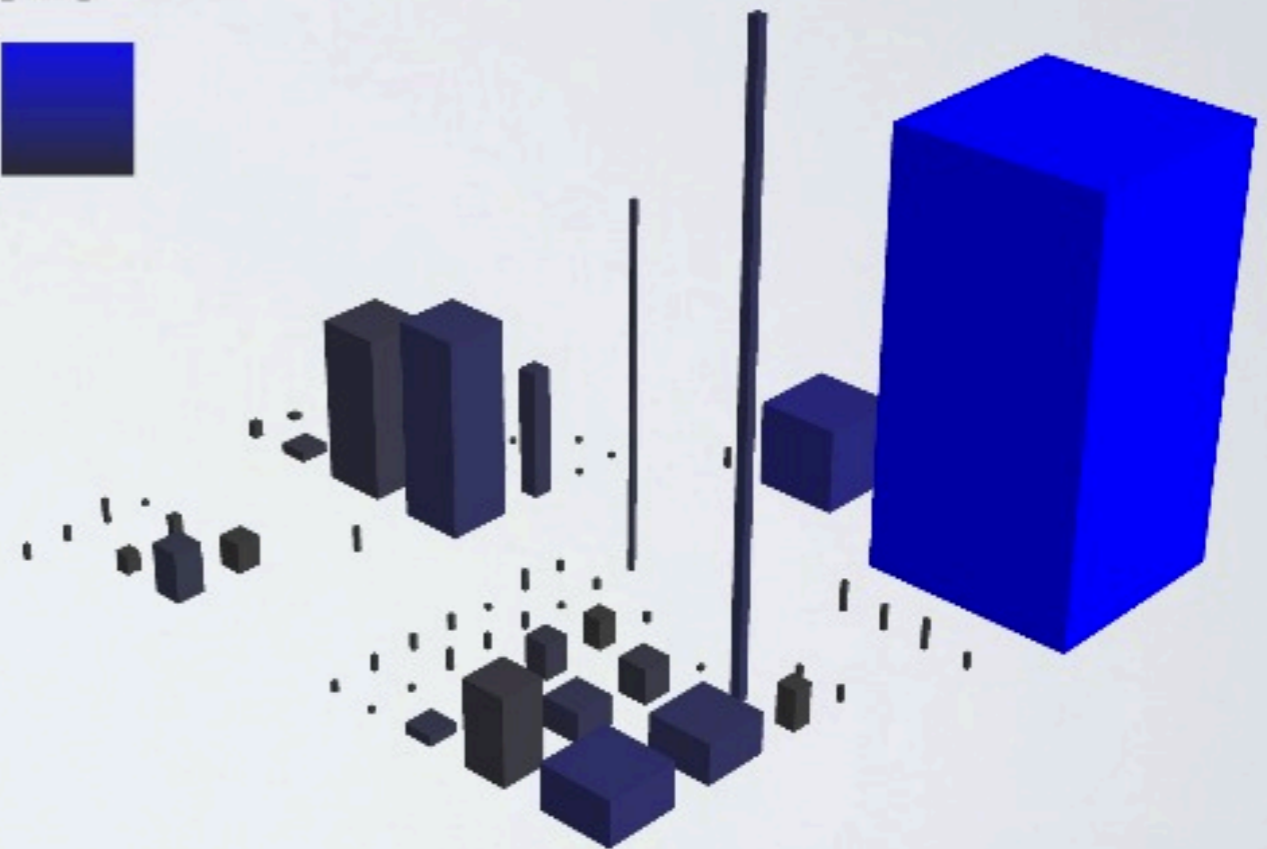
class | building



Our city metaphor

number of methods (NOM)	height
number of attributes (NOA)	base size
number of lines of code (LOC)	color 

class | **building**



Our city metaphor

number of methods (NOM)

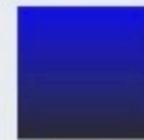
number of attributes (NOA)

number of lines of code (LOC)

height

base size

color

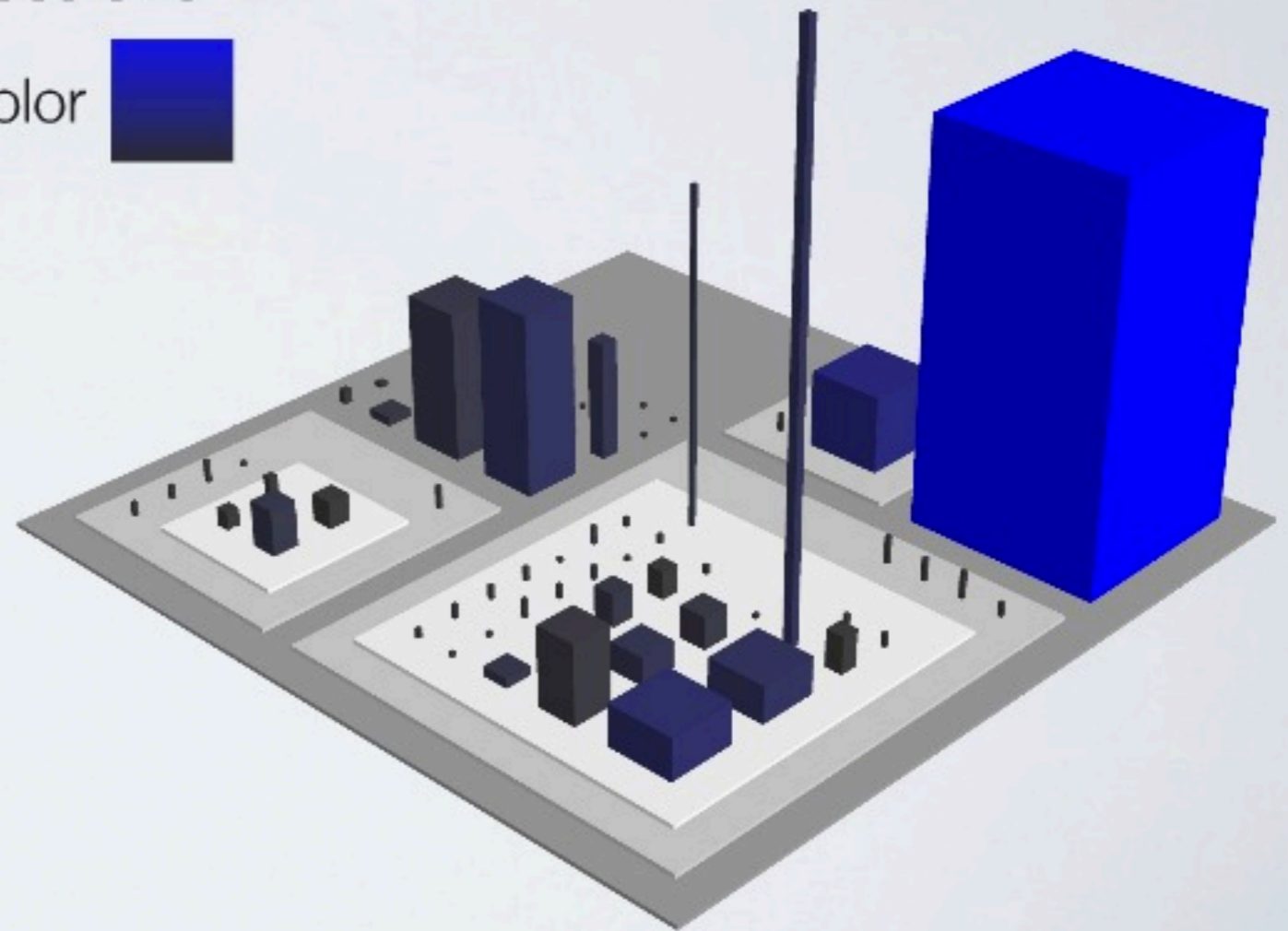


class

building

package

district

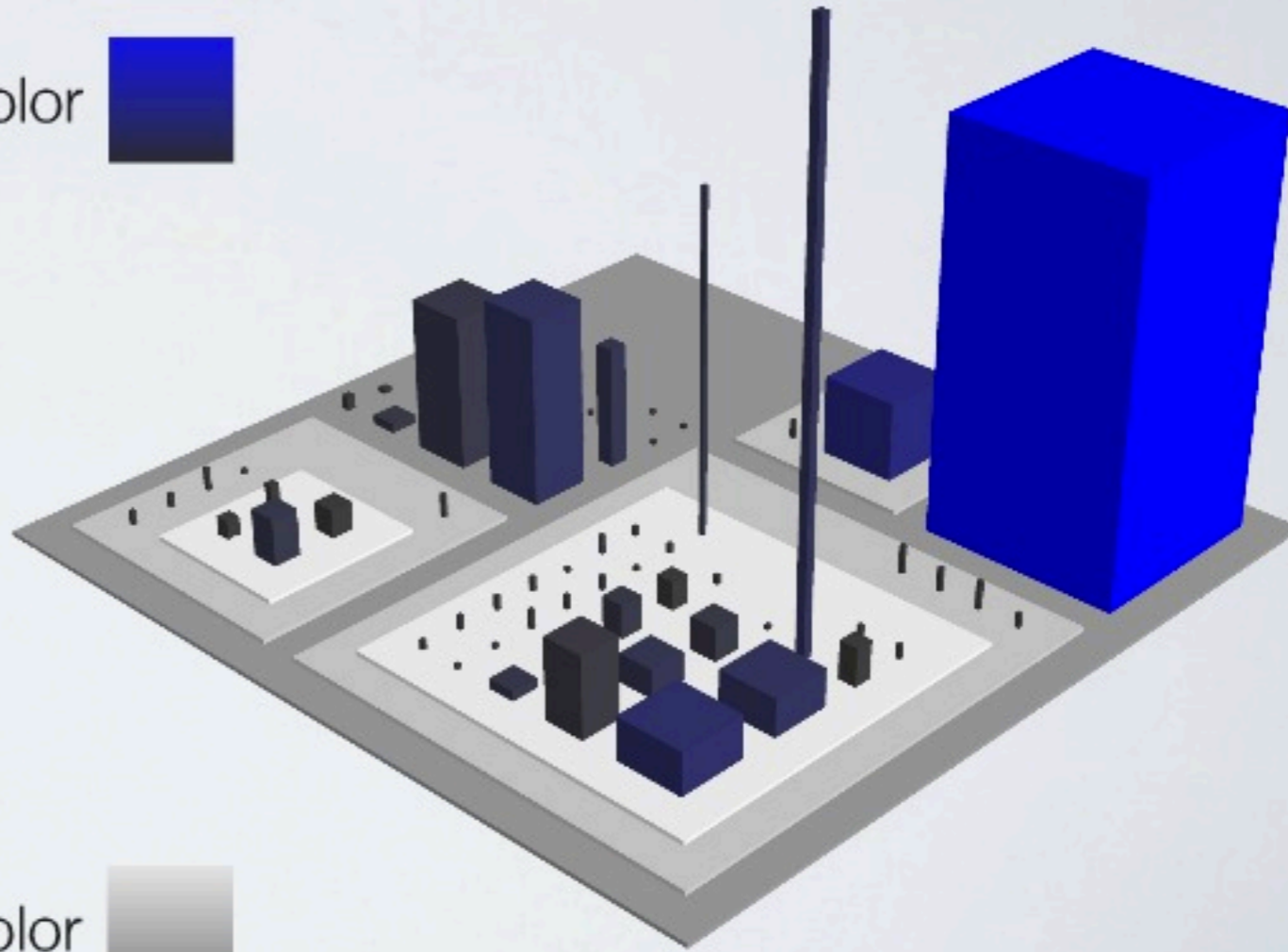


Our city metaphor

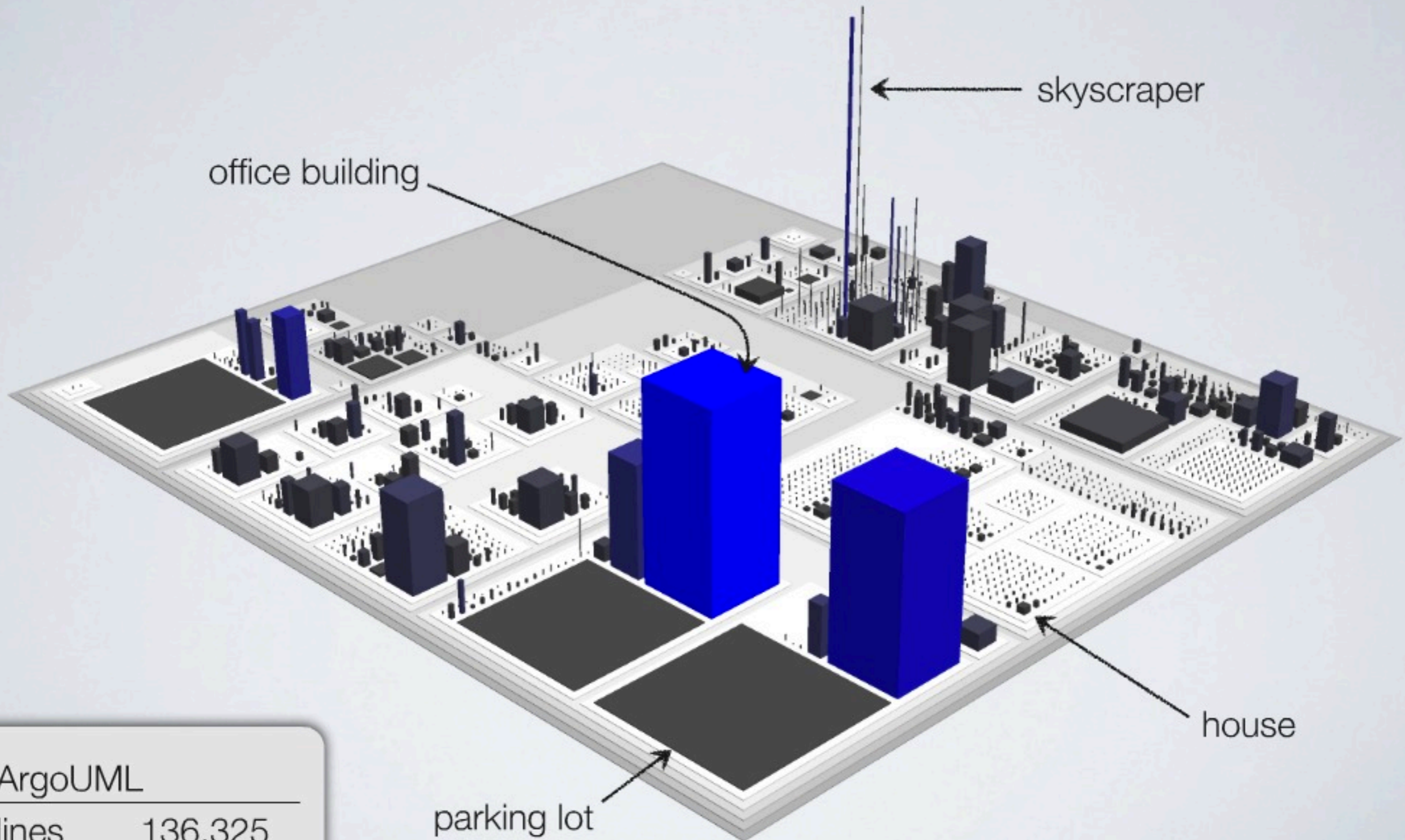
number of methods (NOM) | height
number of attributes (NOA) | base size
number of lines of code (LOC) | color

class | building
package | district

nesting level | color



Code city visualization

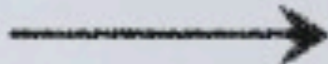


ArgoUML

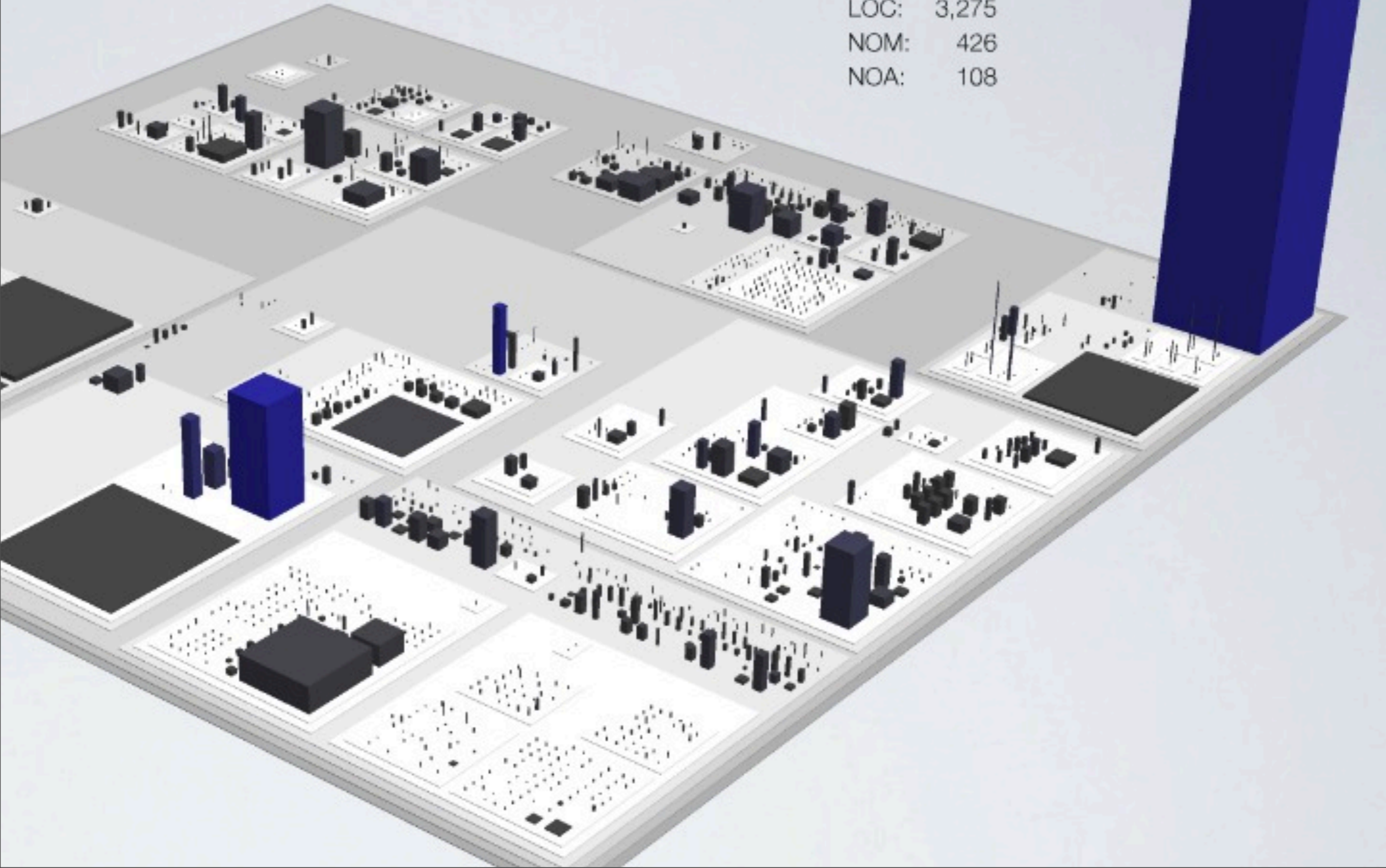
lines 136,325

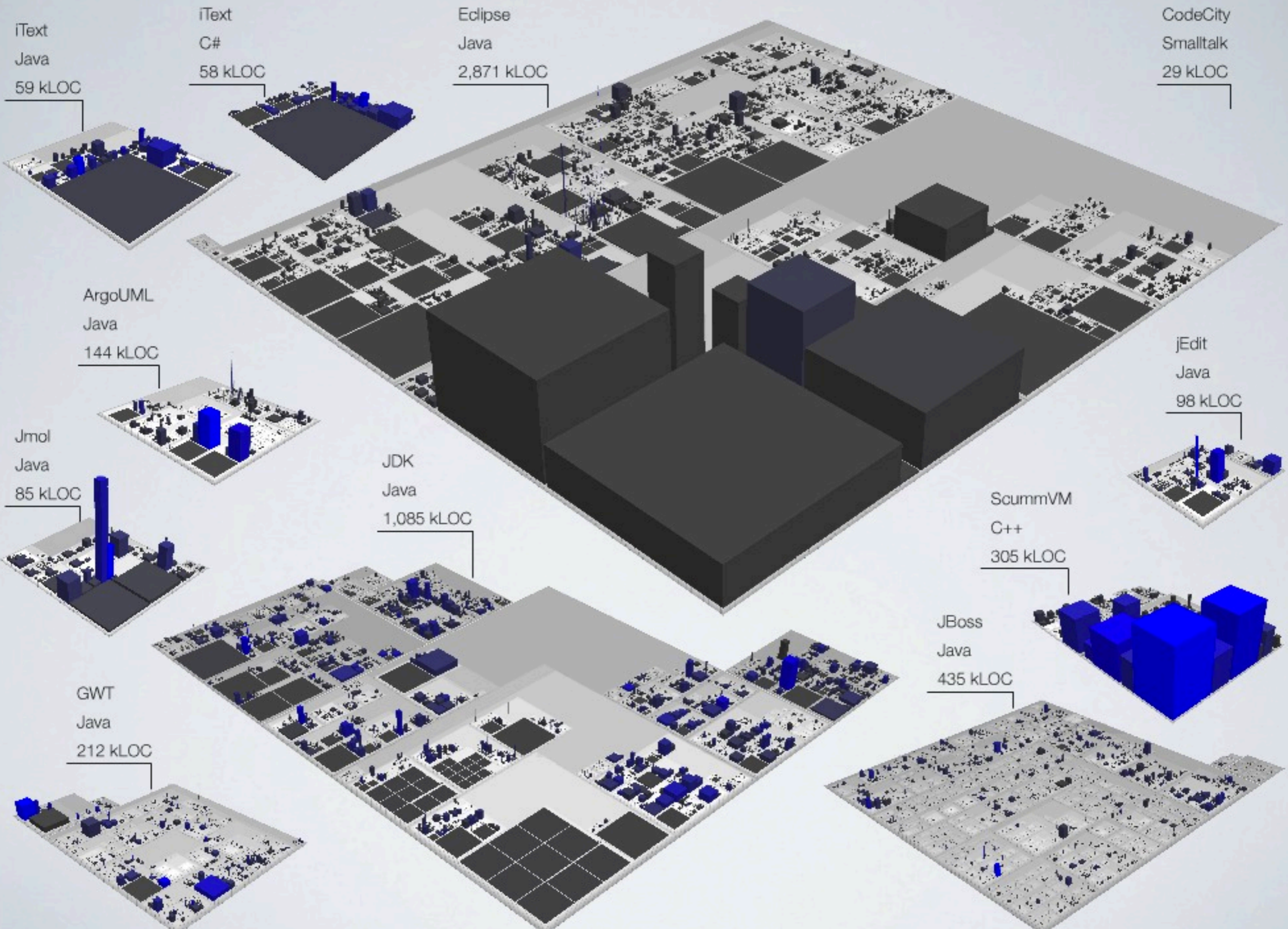
© Richard Wettel, University of Lugano

ModelFacade



model
LOC: 3,275
NOM: 426
NOA: 108





© Richard Wettel, University of Lugano

ArgoUML 1,776

Brain Class	8
God Class	30
God + Brain Class	6
Data Class	17

FacadeMDRImpl

uml.model.mdr
 NOM 349
 NOA 3

CPPParser

reveng.classfile
 NOM 204
 NOA 85

JavaRecognizer

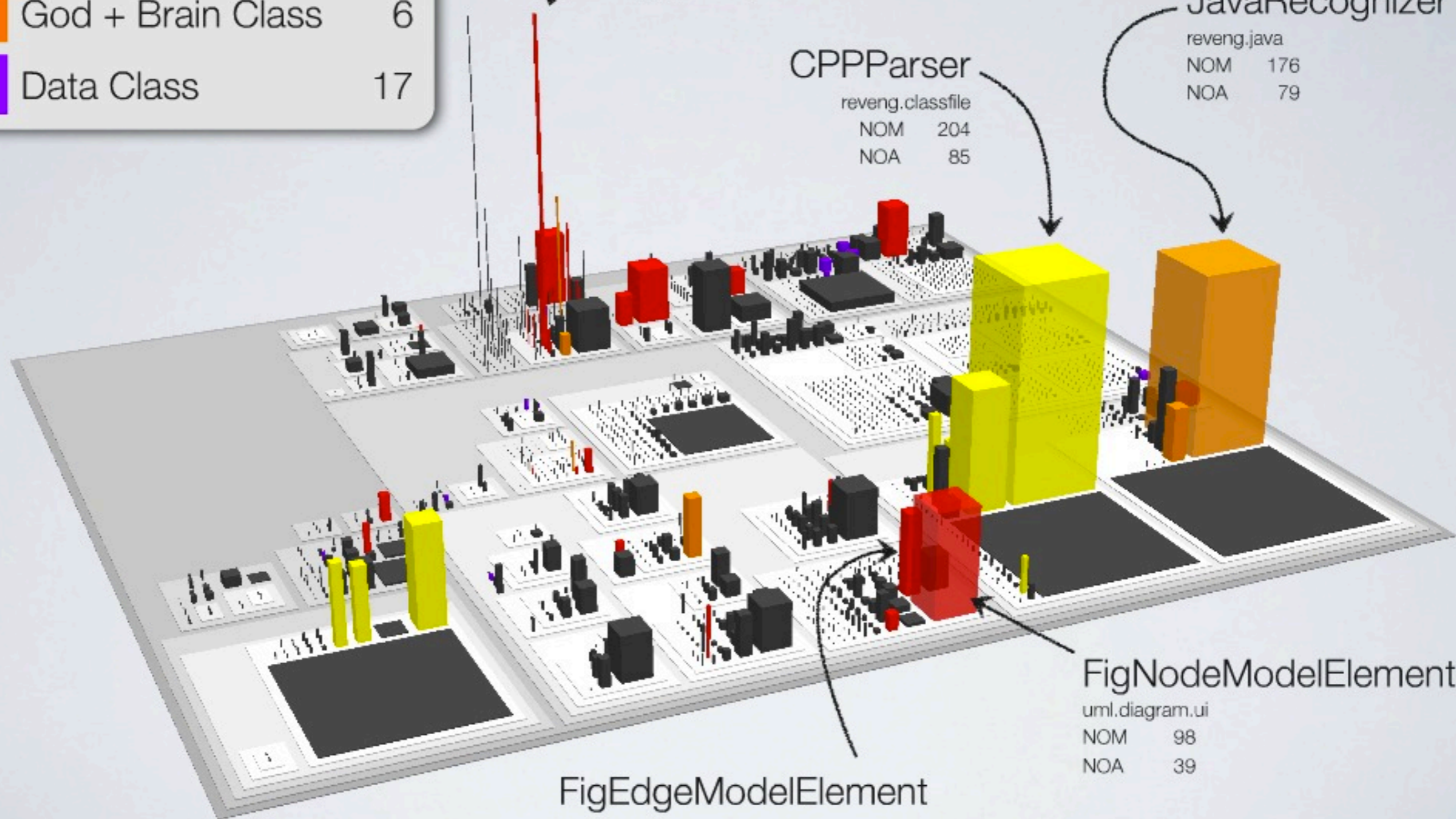
reveng.java
 NOM 176
 NOA 79

FigNodeModelElement

uml.diagram.ui
 NOM 98
 NOA 39

FigEdgeModelElement

uml.diagram.ui
 NOM 73
 NOA 13





In which house do you want to live? - Use *analogies* from *daily life*

CocoViz

<http://seal.ifi.uzh.ch/cocoviz>

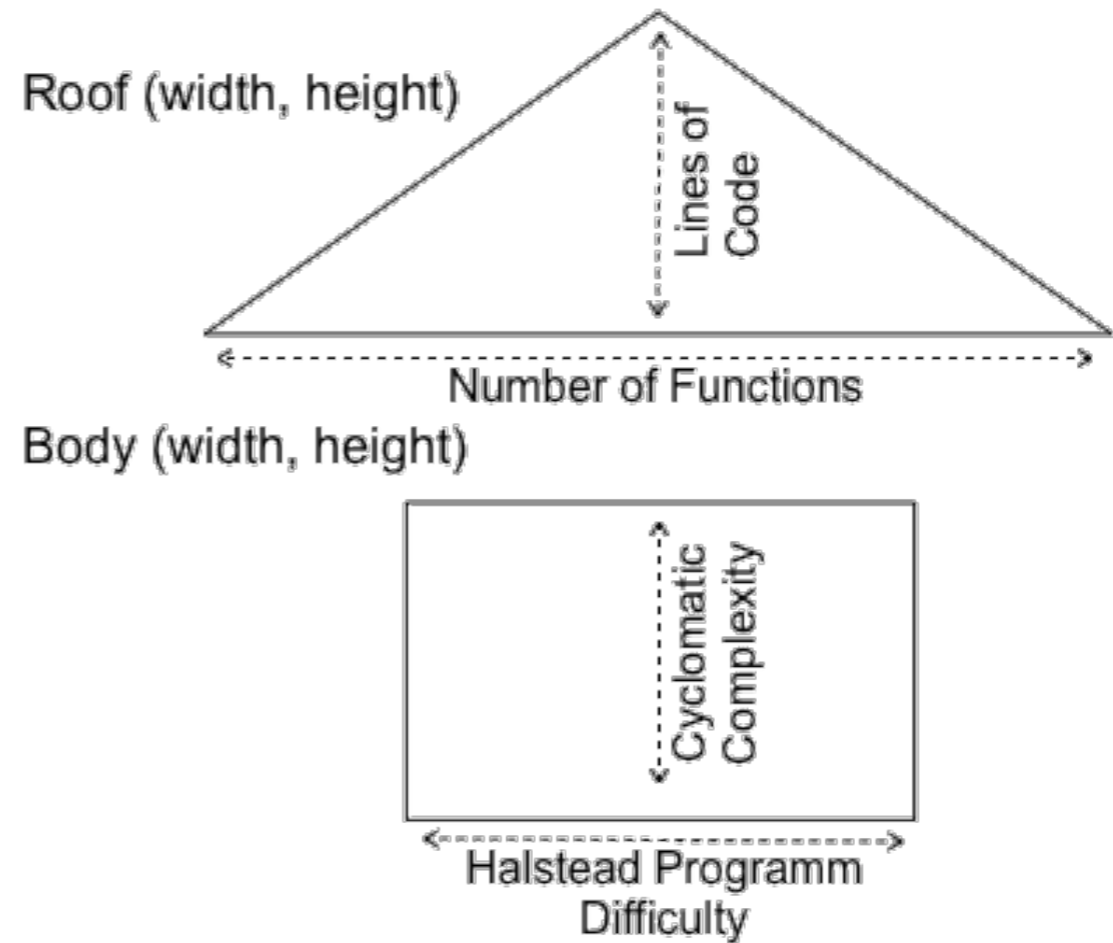
Provides software comprehension support through visualization and audio.

Objects from daily life are used as metaphors to foster an intuitive assessment of a software's structure and quality.

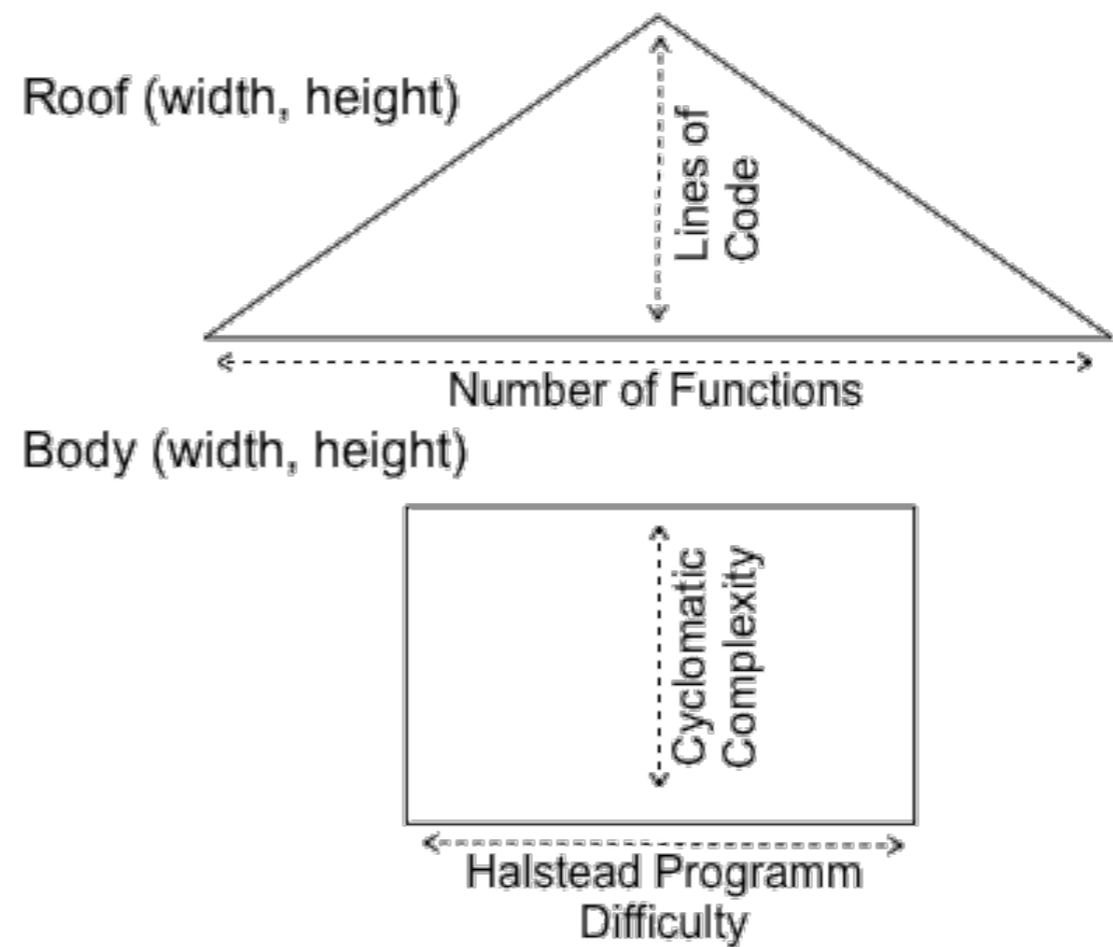
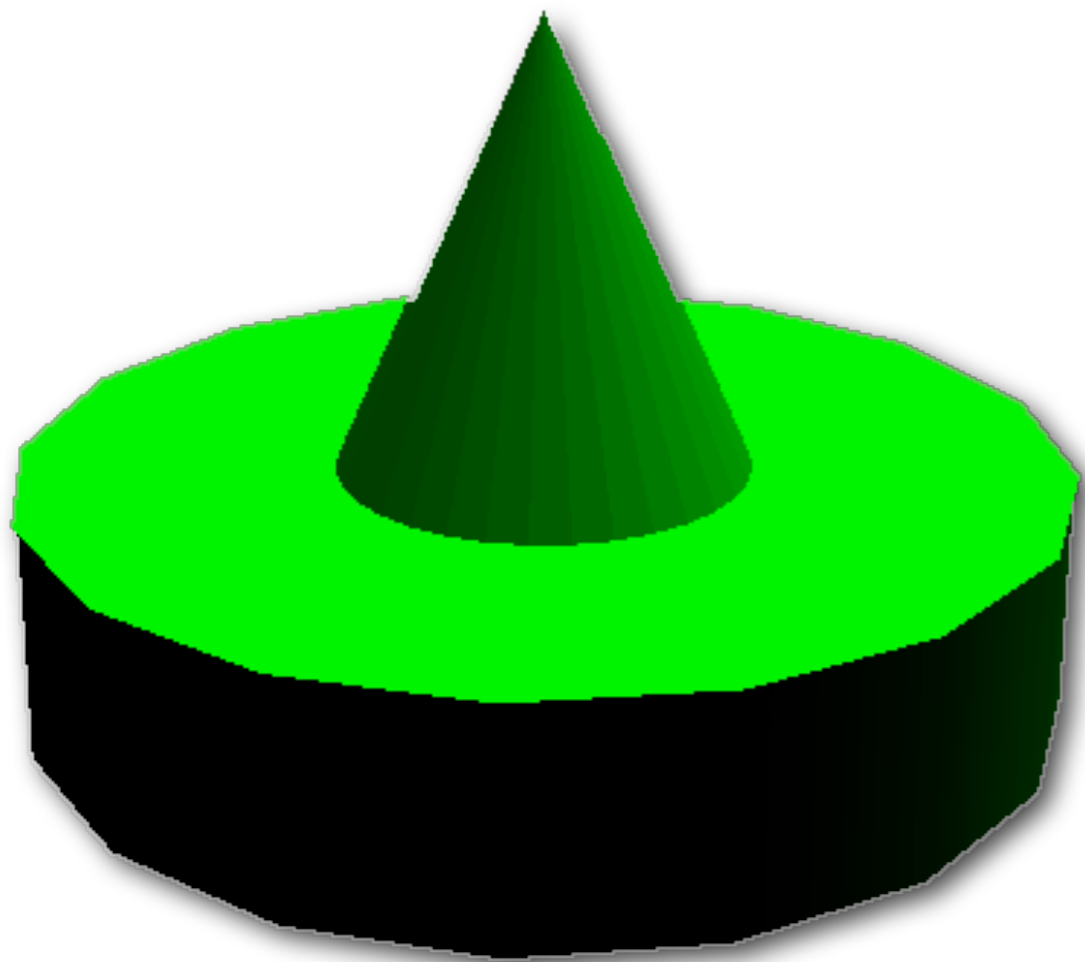
Incorporates static source code metrics and evolutionary aspects.



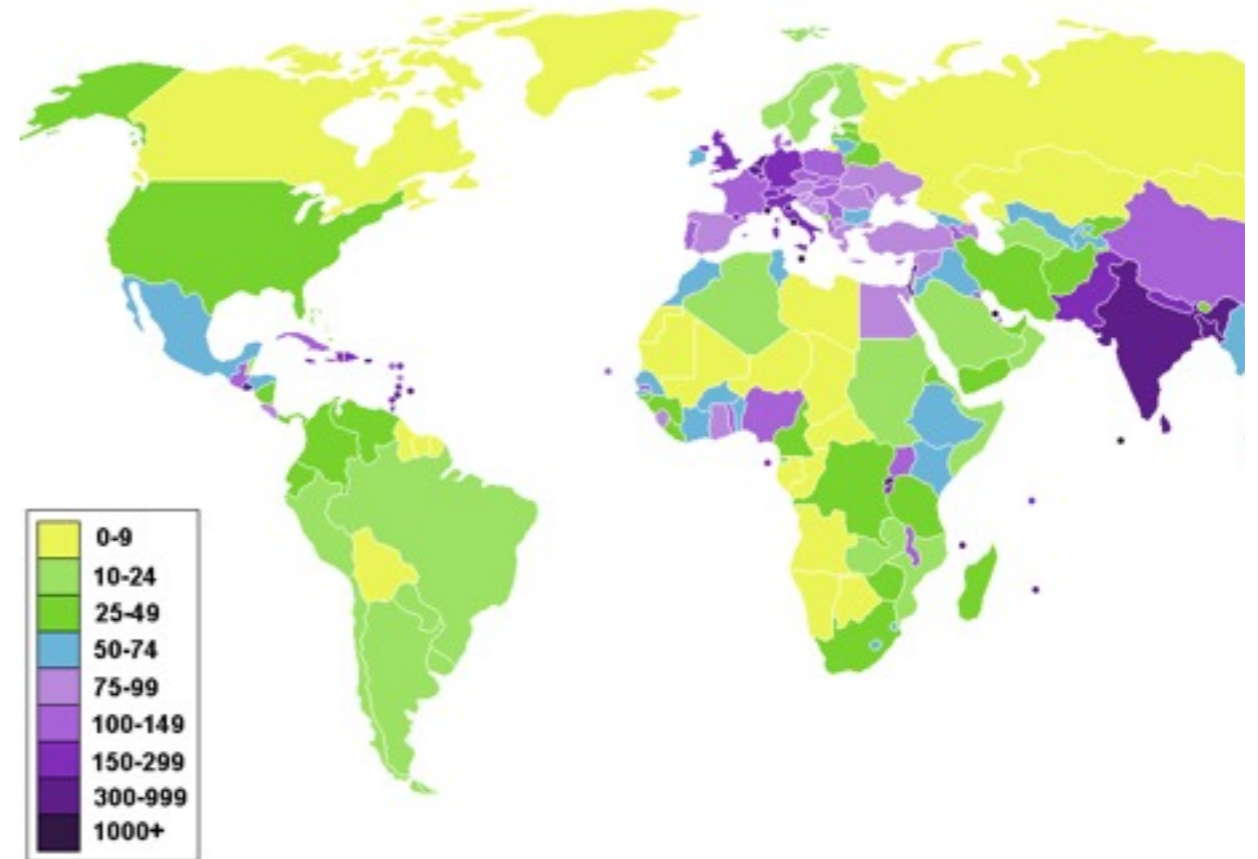
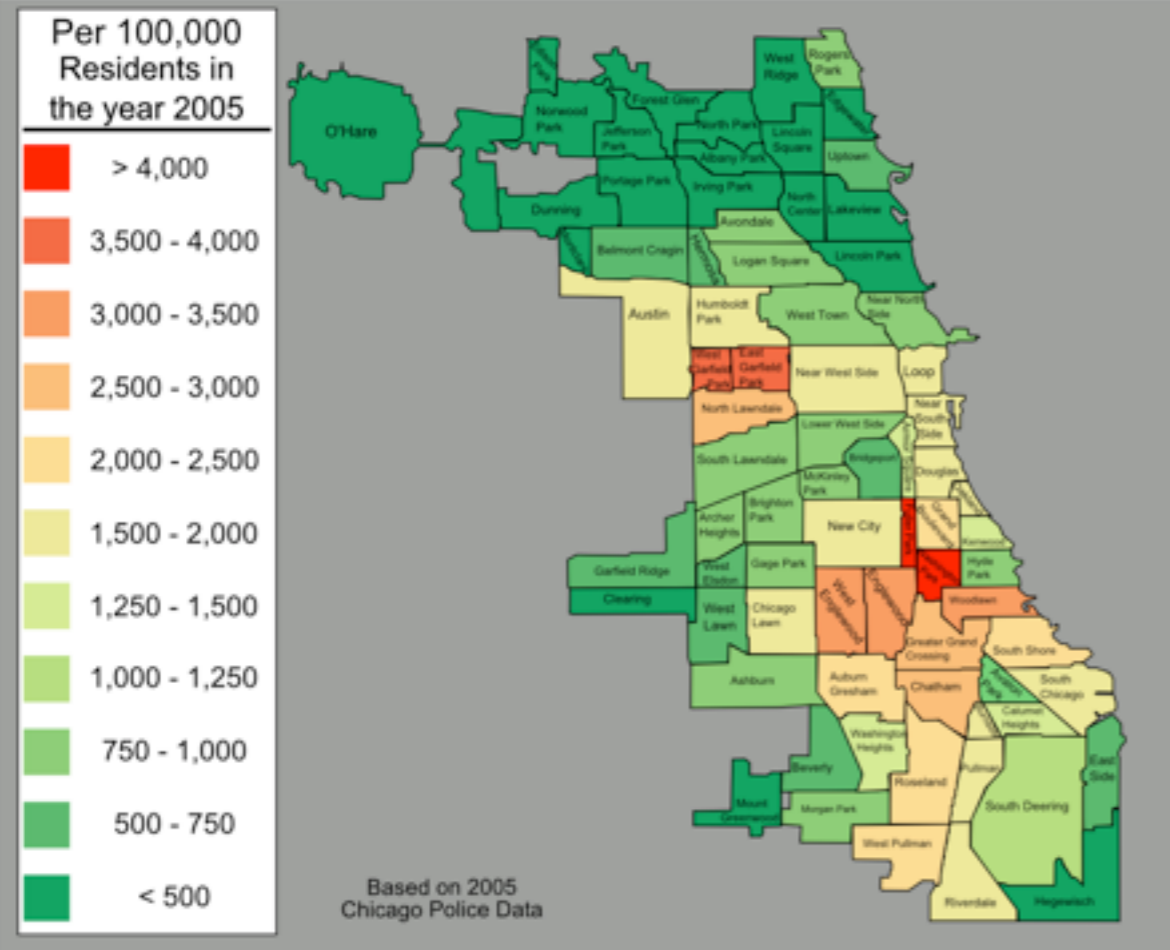
CocoViz: Cognitive Glyphs



CocoViz: Cognitive Glyphs



Violent Crimes by Neighborhood



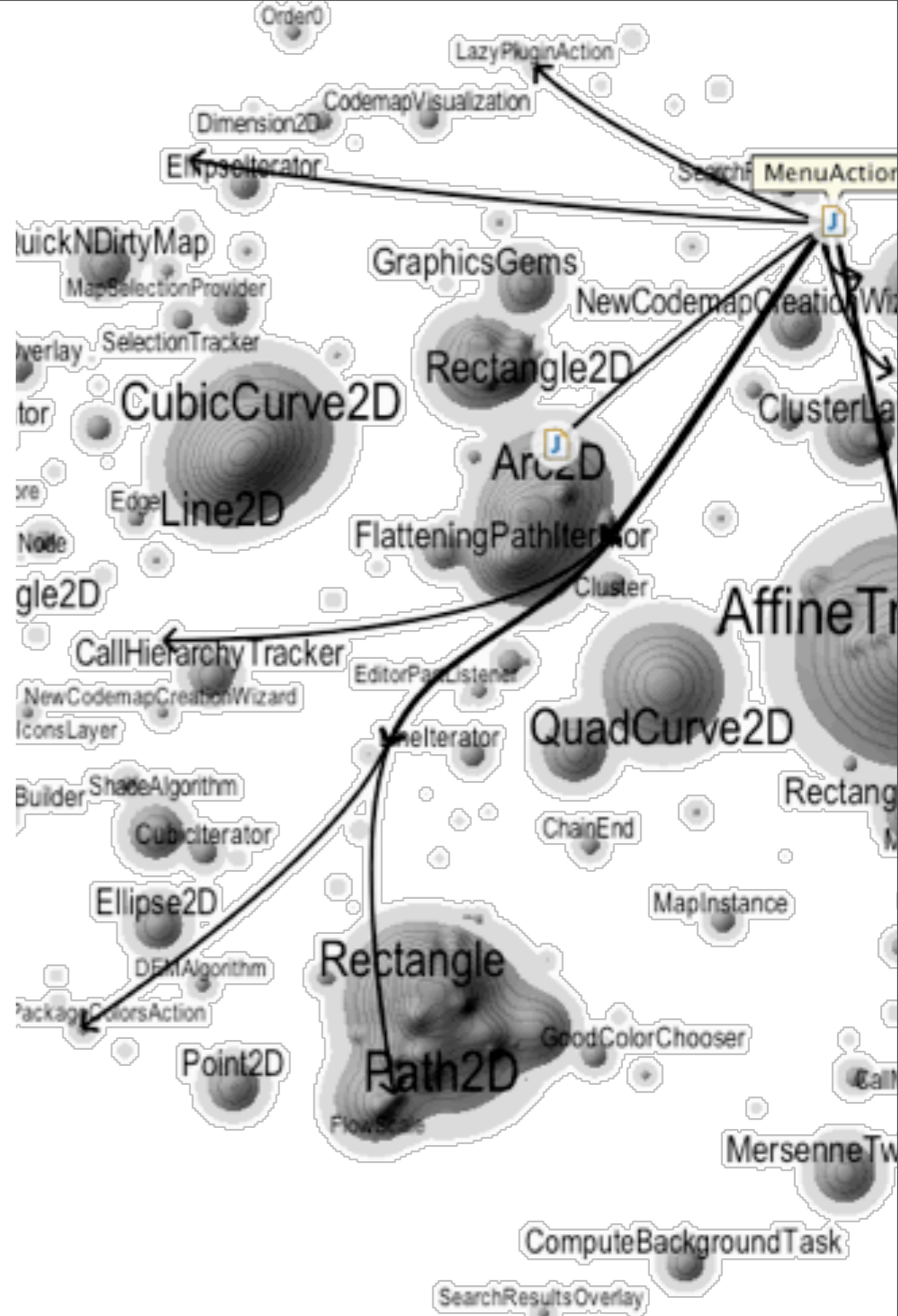
Use methods from GIS applications form SW Visualization

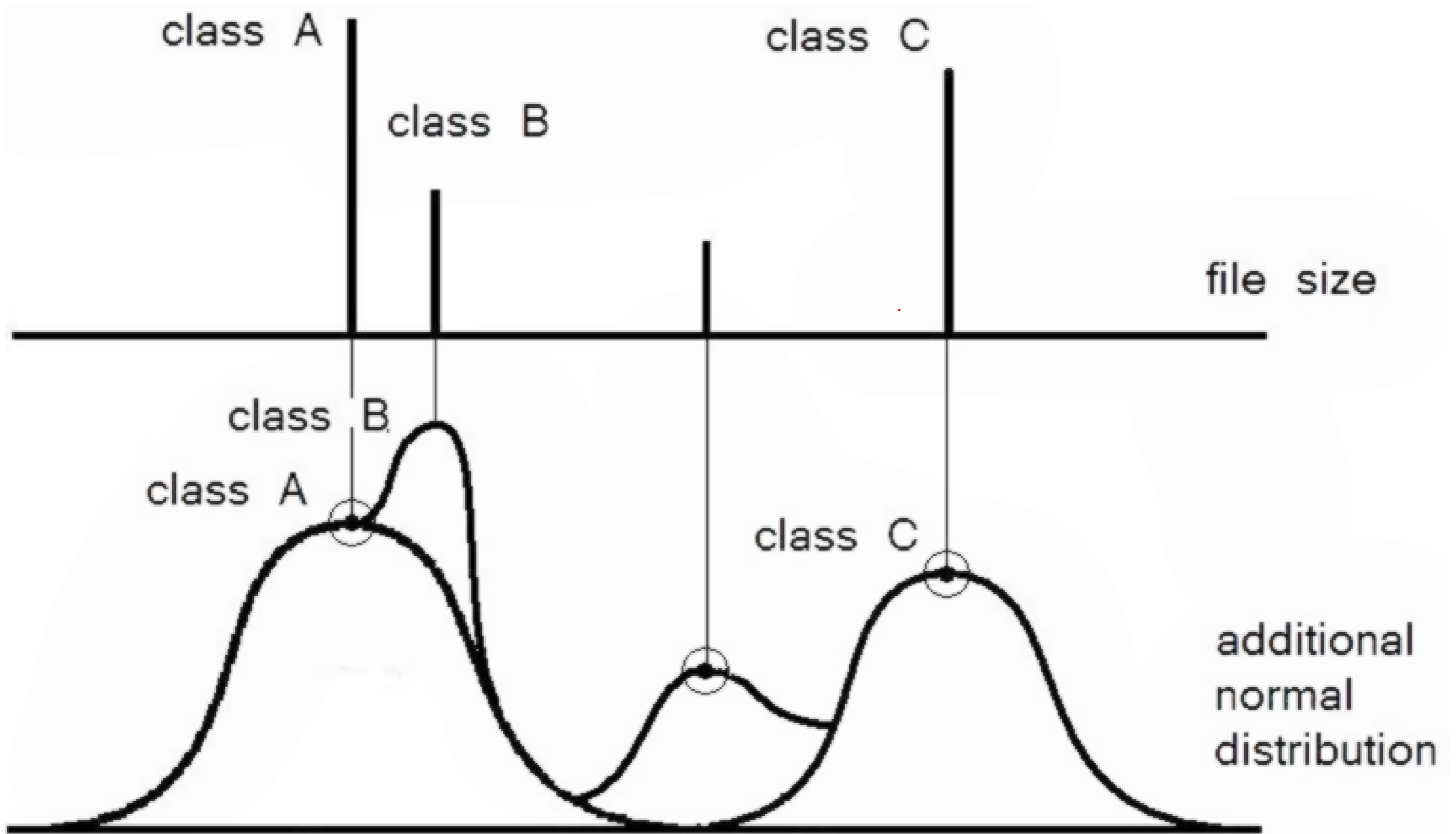
Software Cartography

<http://scg.unibe.ch/codemap>

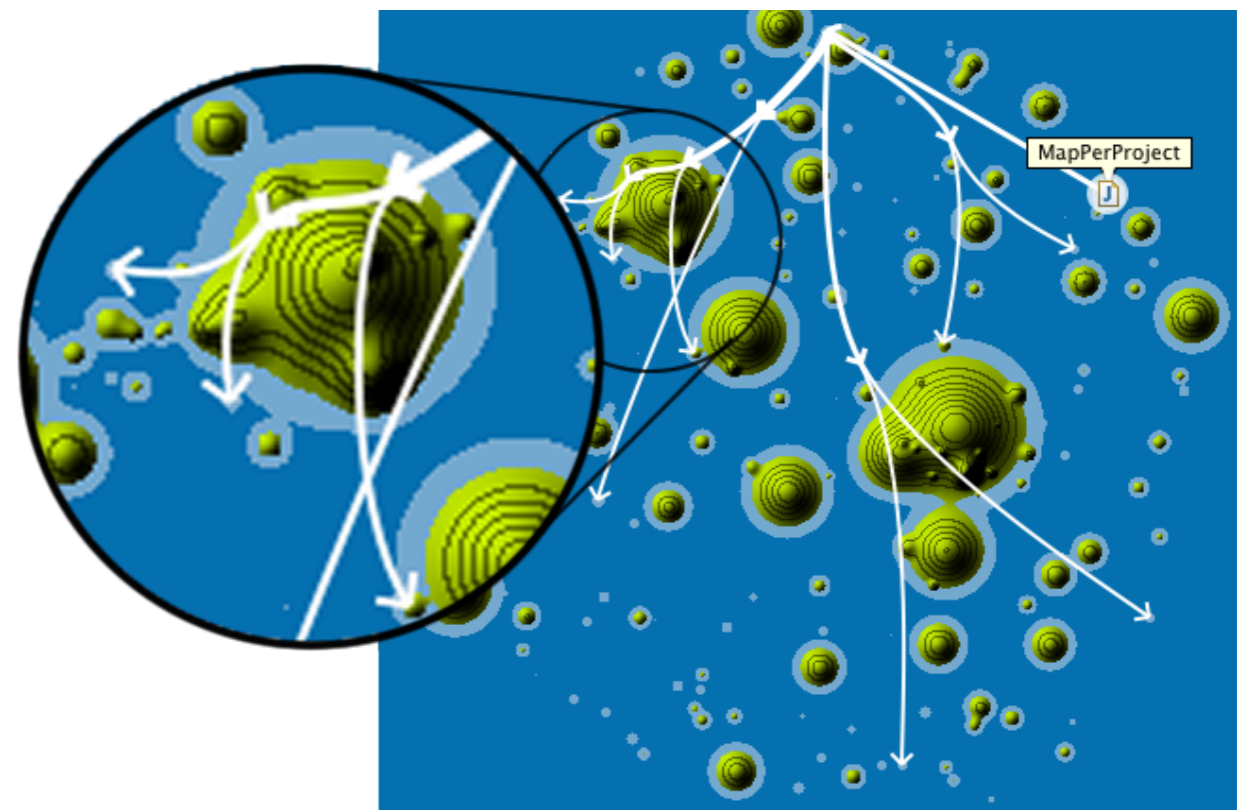
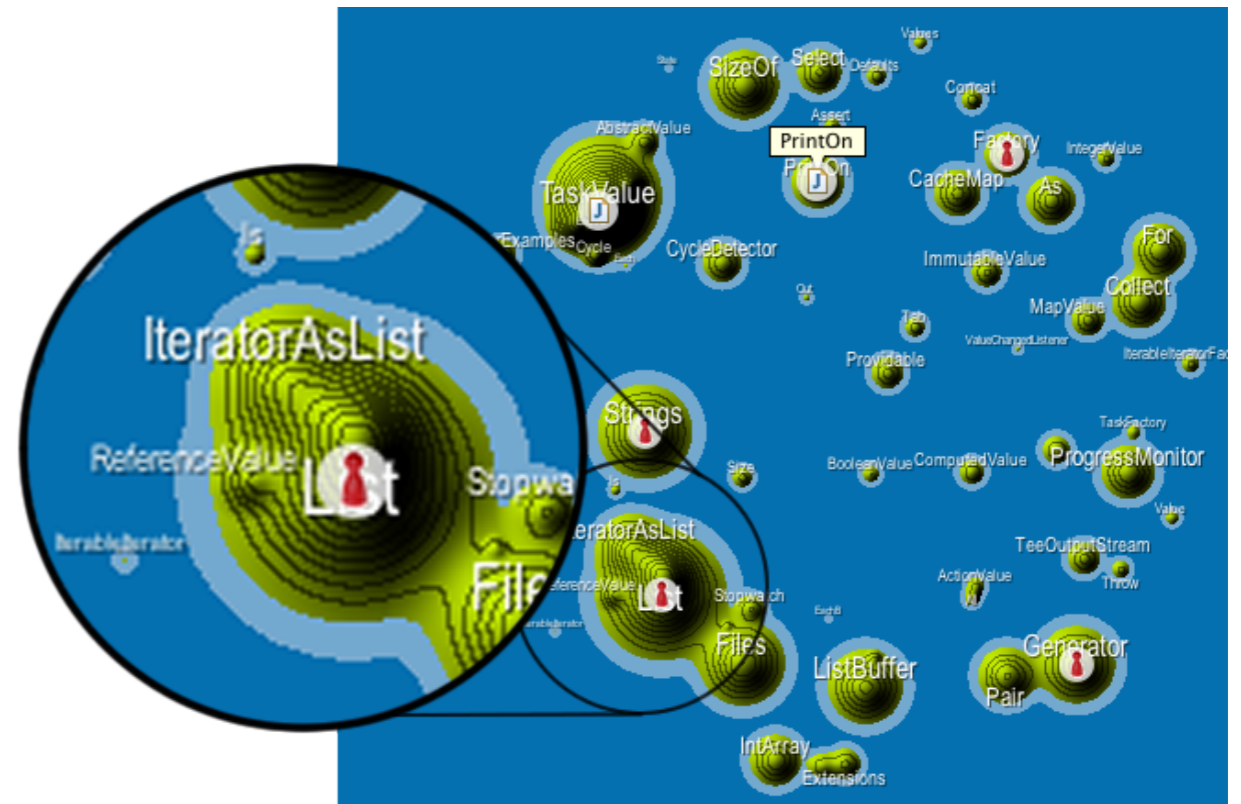
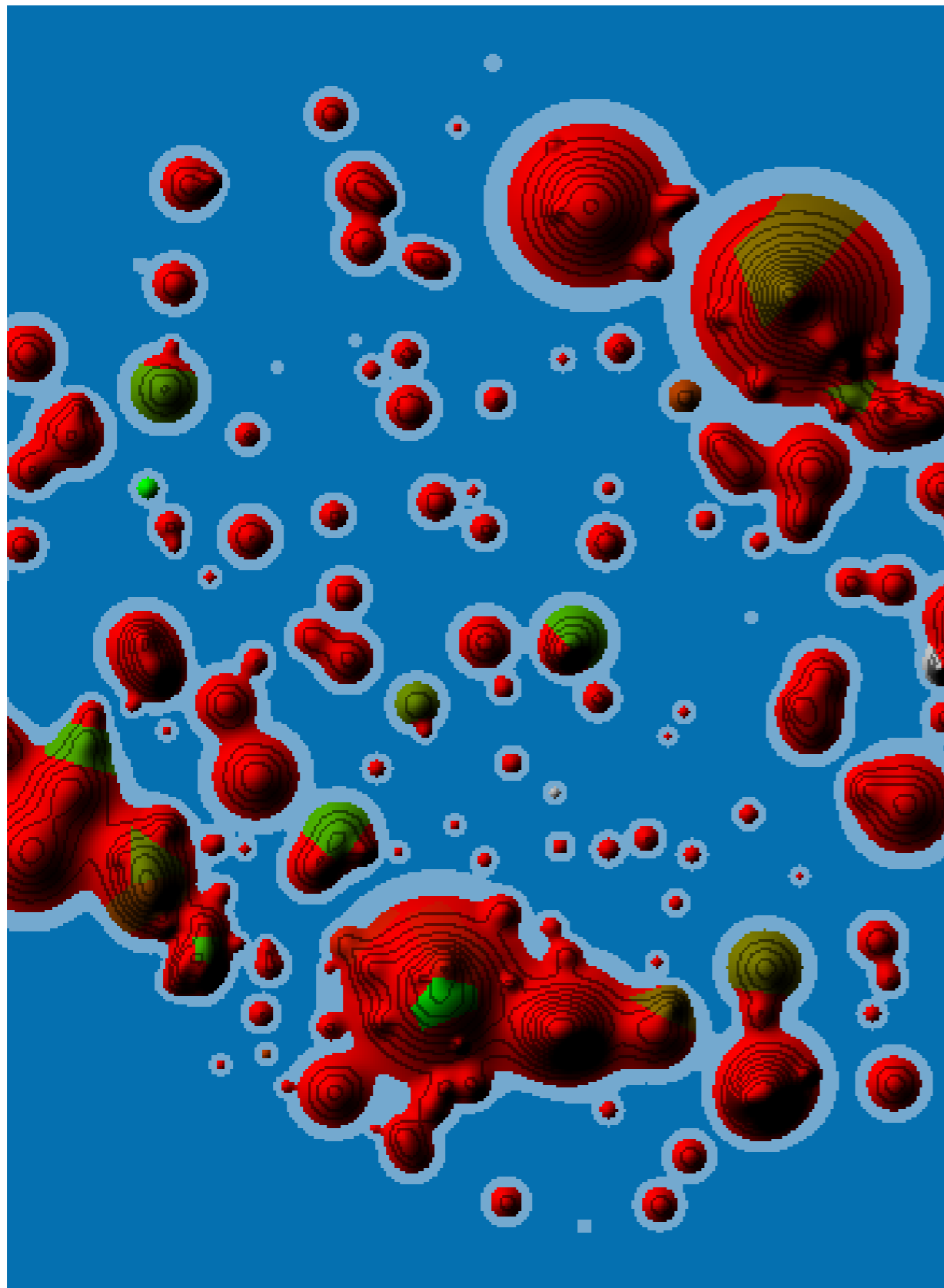
Codemaps use the same visual language as cartographic visualizations found in an atlas.

- Population density
- Industry sectors
- weather forecast
- birth rate
- flow of trade
- ...



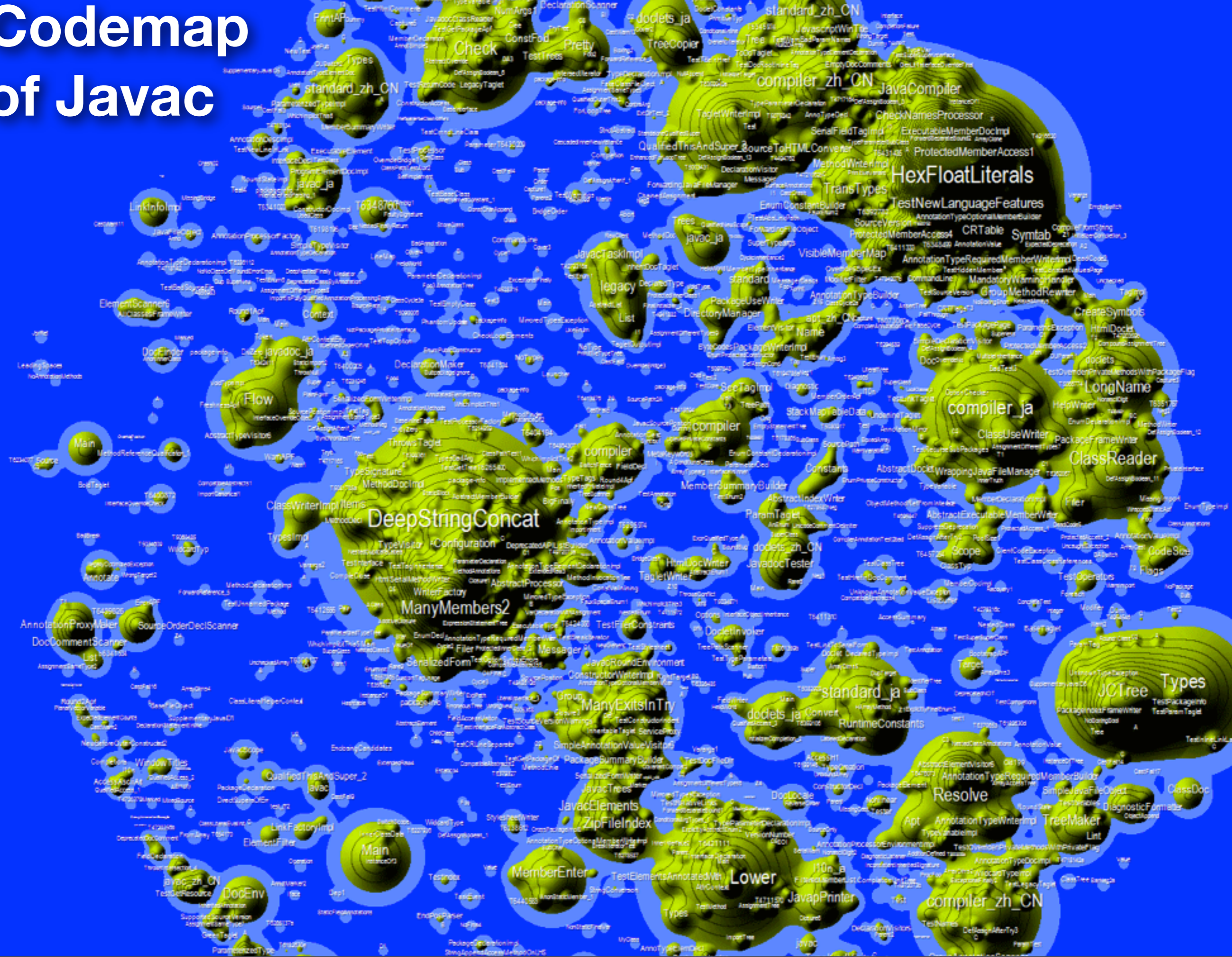


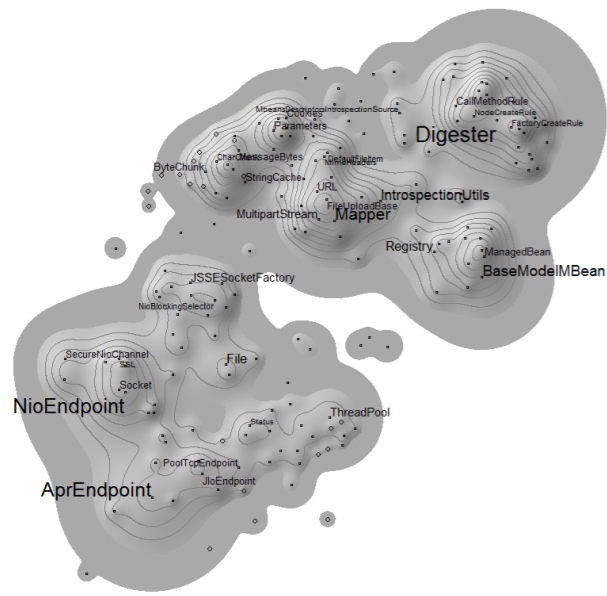
Digital elevation model of Codemaps



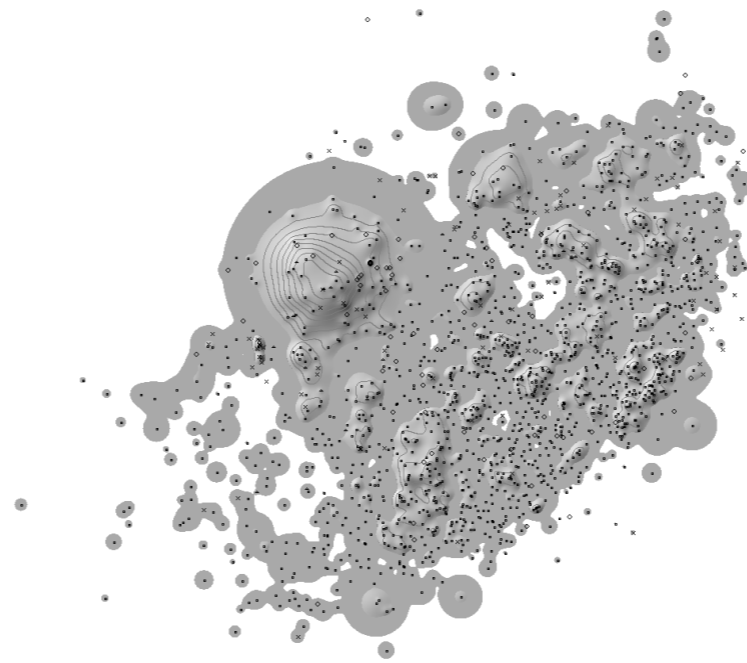
Codemap: Improving the Mental Model of Software Developers through Cartographic Visualization. David Erni. Master's Thesis, University of Bern

Codemap of Javac

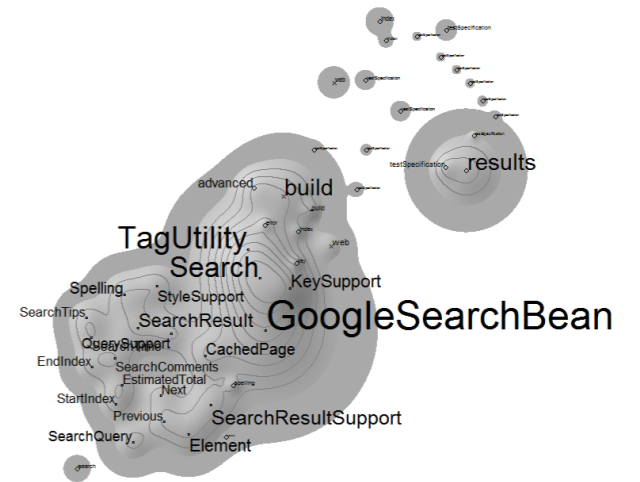




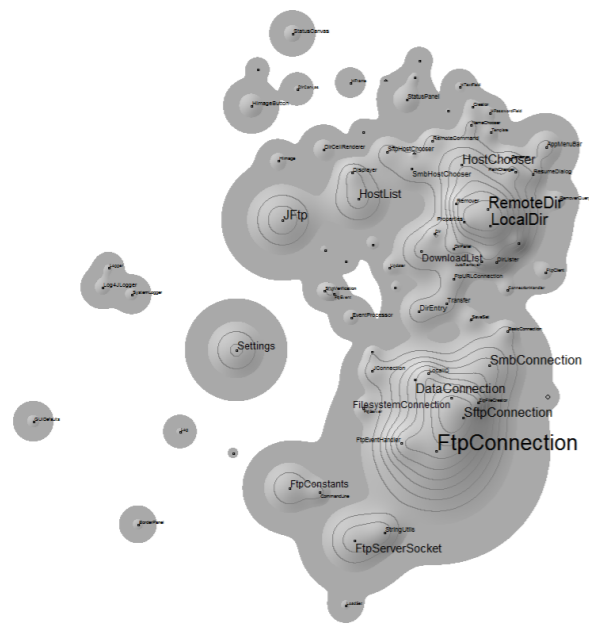
Apache Tomcat



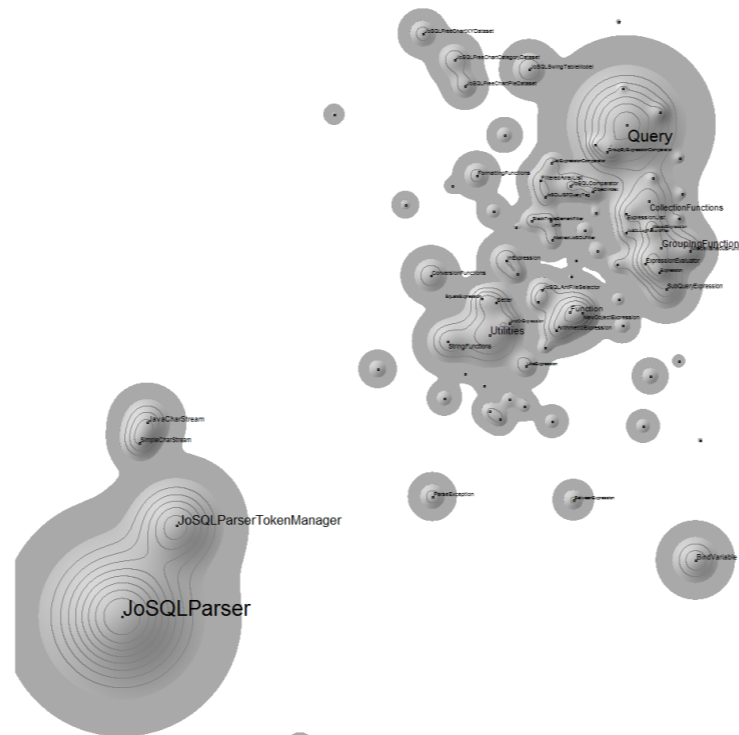
Columba



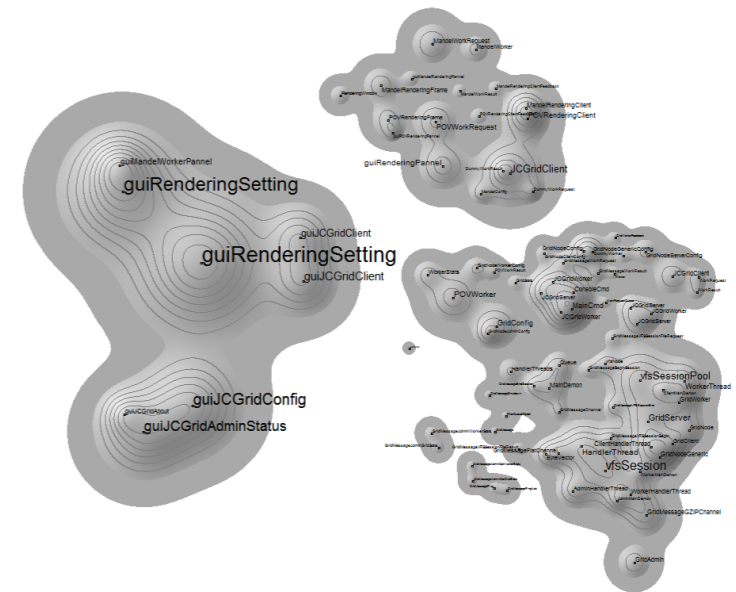
Google Taglib



JFtp



JoSQL



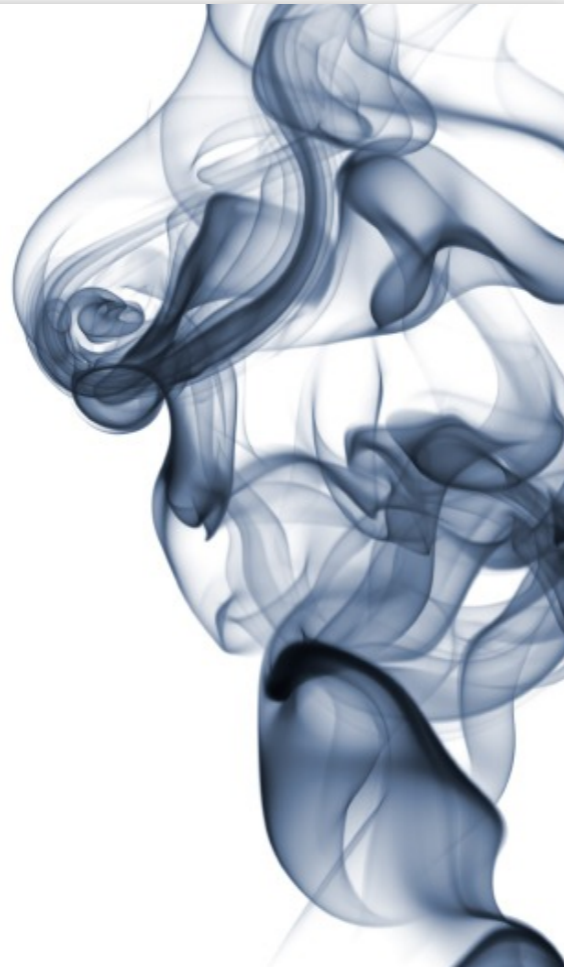
JCGrid

Software Cartography: A Prototype for Thematic Software Maps. Peter Loretan, Diploma thesis, University of Bern

Ball's Dilemma

*Software is **intangible**,
having no physical
shape or size.*

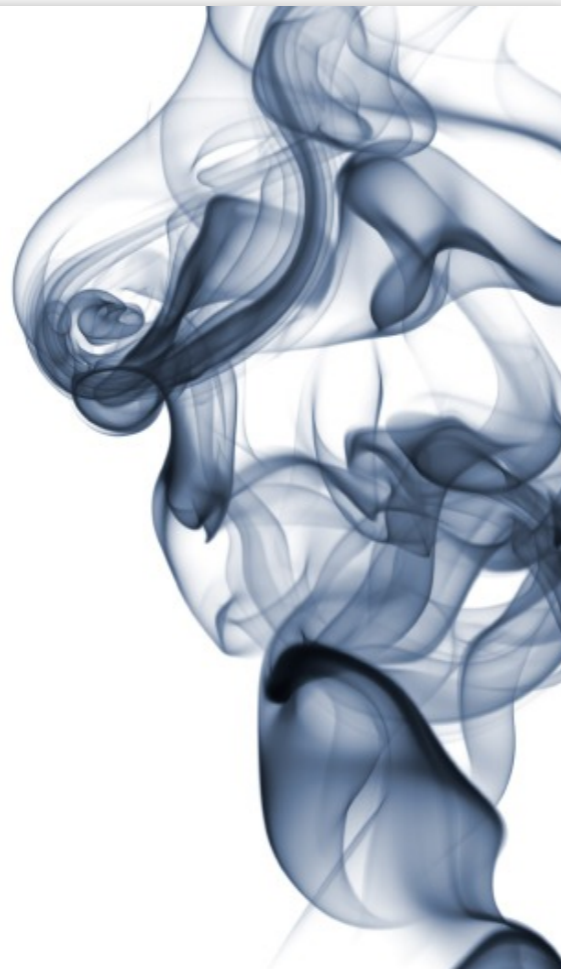
Thomas Ball



Ball's Dilemma

Software is *intangible*,
having no physical
shape or size.

Thomas Ball



Metrics

Metrics are a mapping of a particular characteristic of a measured entity to a numerical value.

They can be used to quantify aspects of quality.

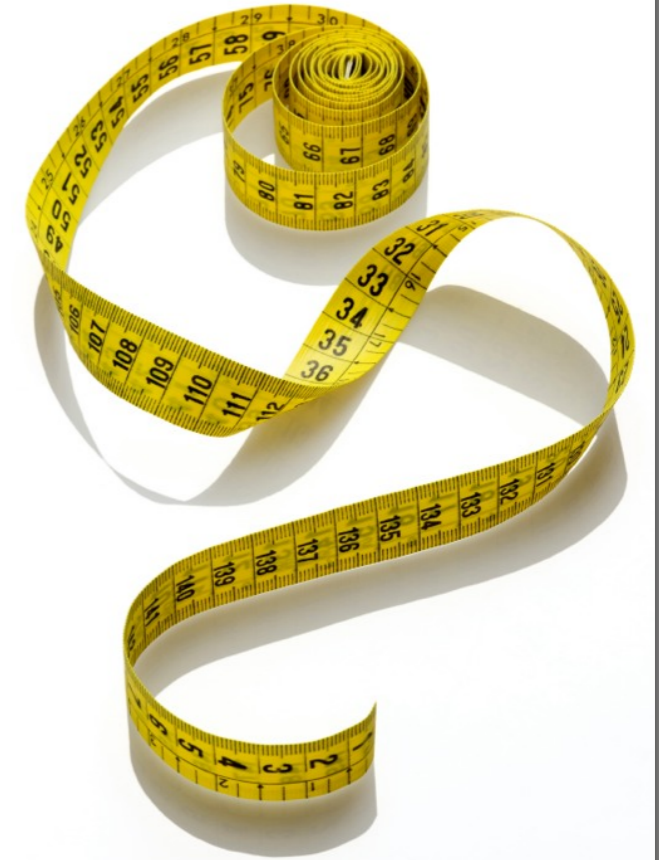
Measuring software is cheap and can be automated.

Numbers are just numbers; don't trust them.

They capture symptoms, not causes.

Hard for developers to deal with them.

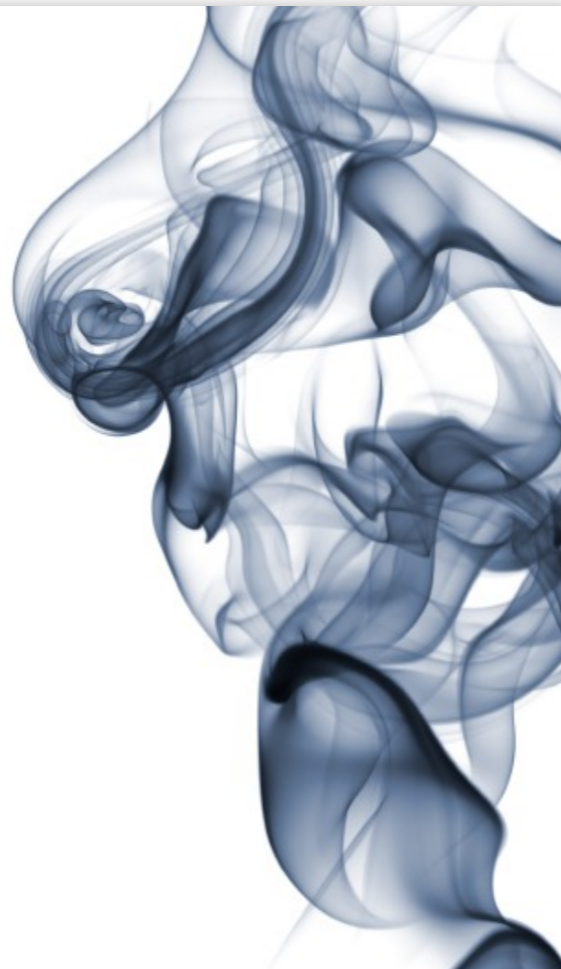
Inflation of measurements



Ball's Dilemma

Software is *intangible*,
having no physical
shape or size.

Thomas Ball



Metrics

Metrics are a mapping of a particular
characteristic of a measured entity to
a numerical value.

They can be used to quantify aspects
of quality.

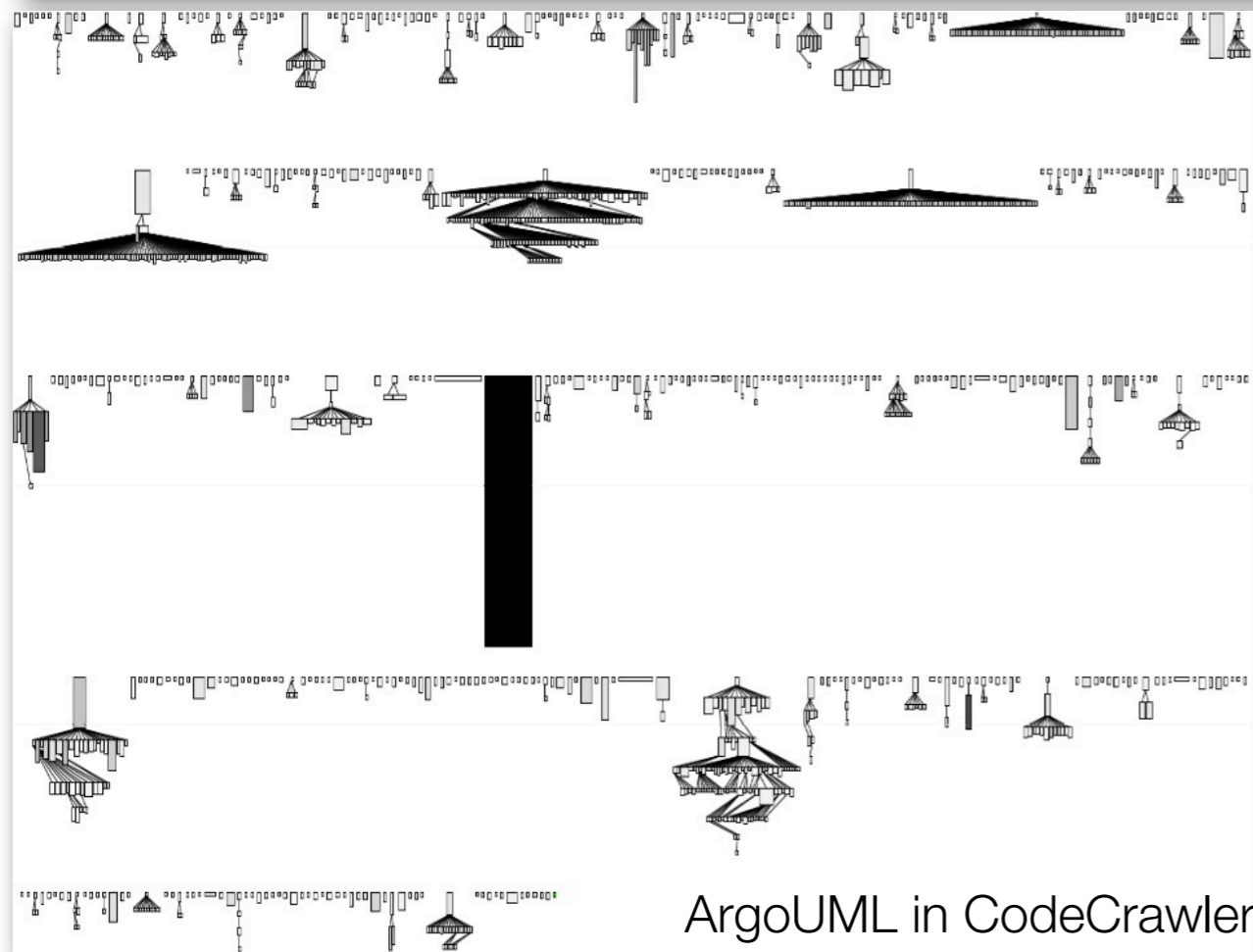
Measuring software is cheap and can
be automated.

Numbers are just numbers; don't
trust them.

They capture symptoms, not causes.

Hard for developers to deal with
them.

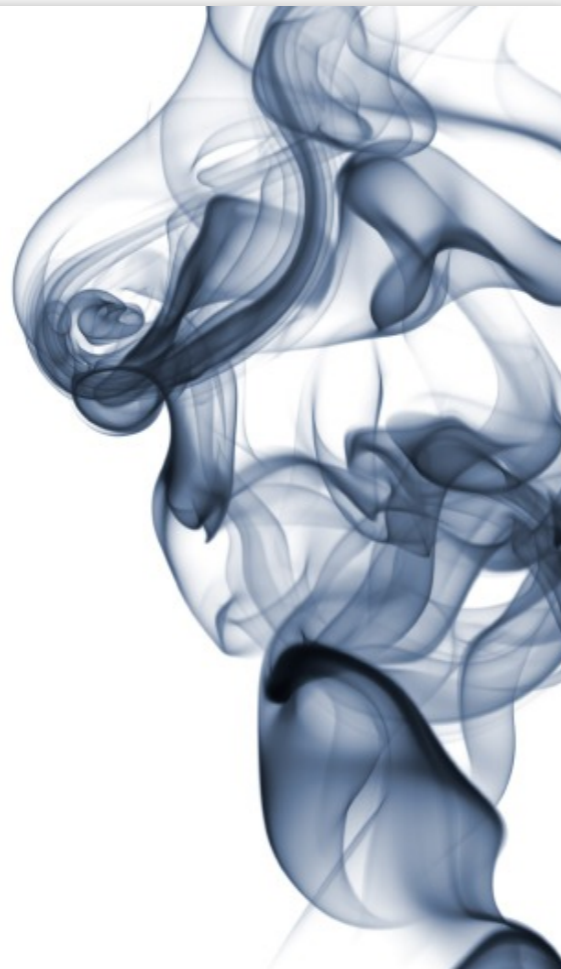
Inflation of measurements



Ball's Dilemma

Software is *intangible*,
having no physical
shape or size.

Thomas Ball



Metrics

Metrics are a mapping of a particular
characteristic of a measured entity to
a numerical value.

They can be used to quantify aspects
of quality.

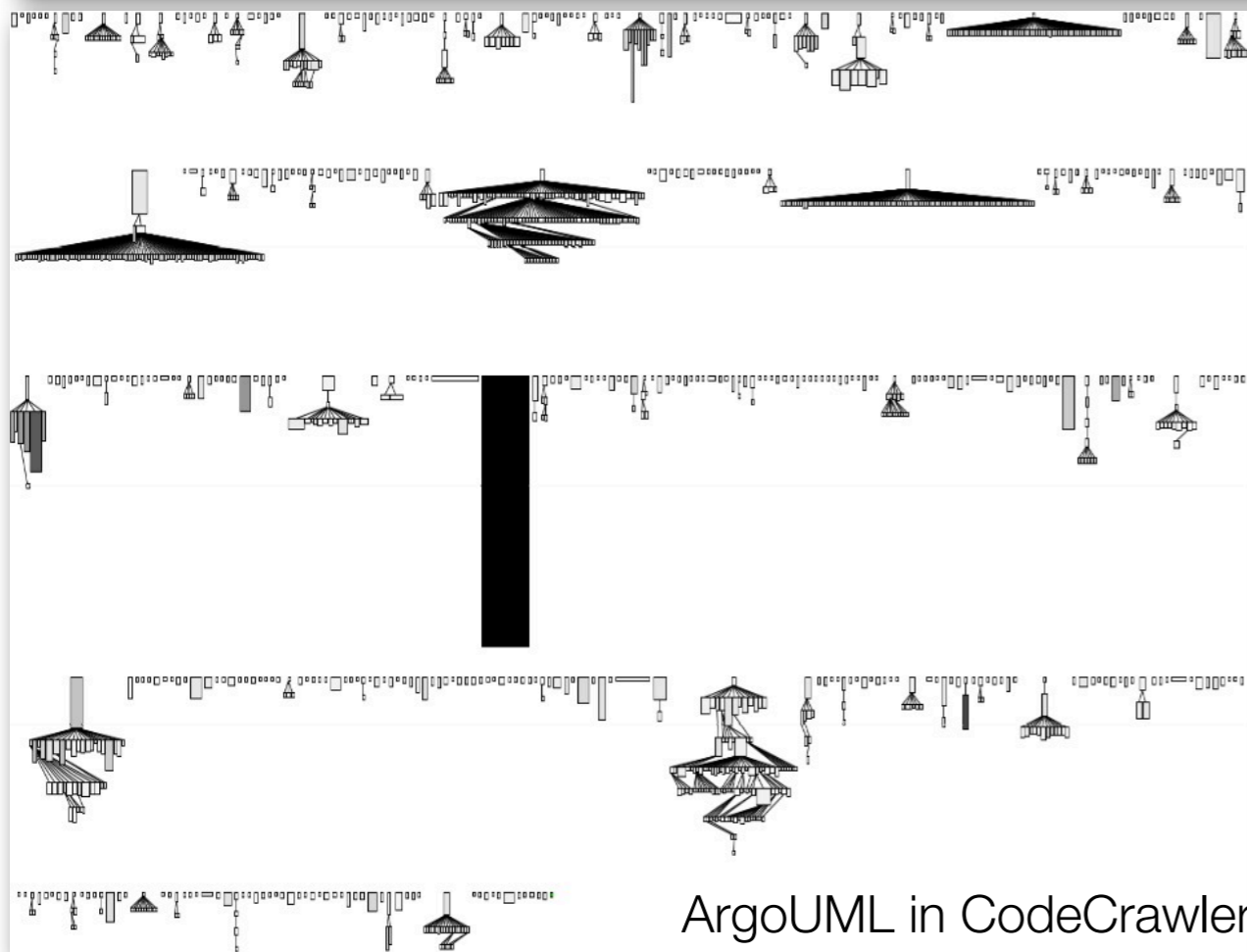
Measuring software is cheap and can
be automated.

Numbers are just numbers; don't
trust them.

They capture symptoms, not causes.

Hard for developers to deal with
them.

Inflation of measurements



About Metaphors

Software Visualization++