

5.– 8. September 2011
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

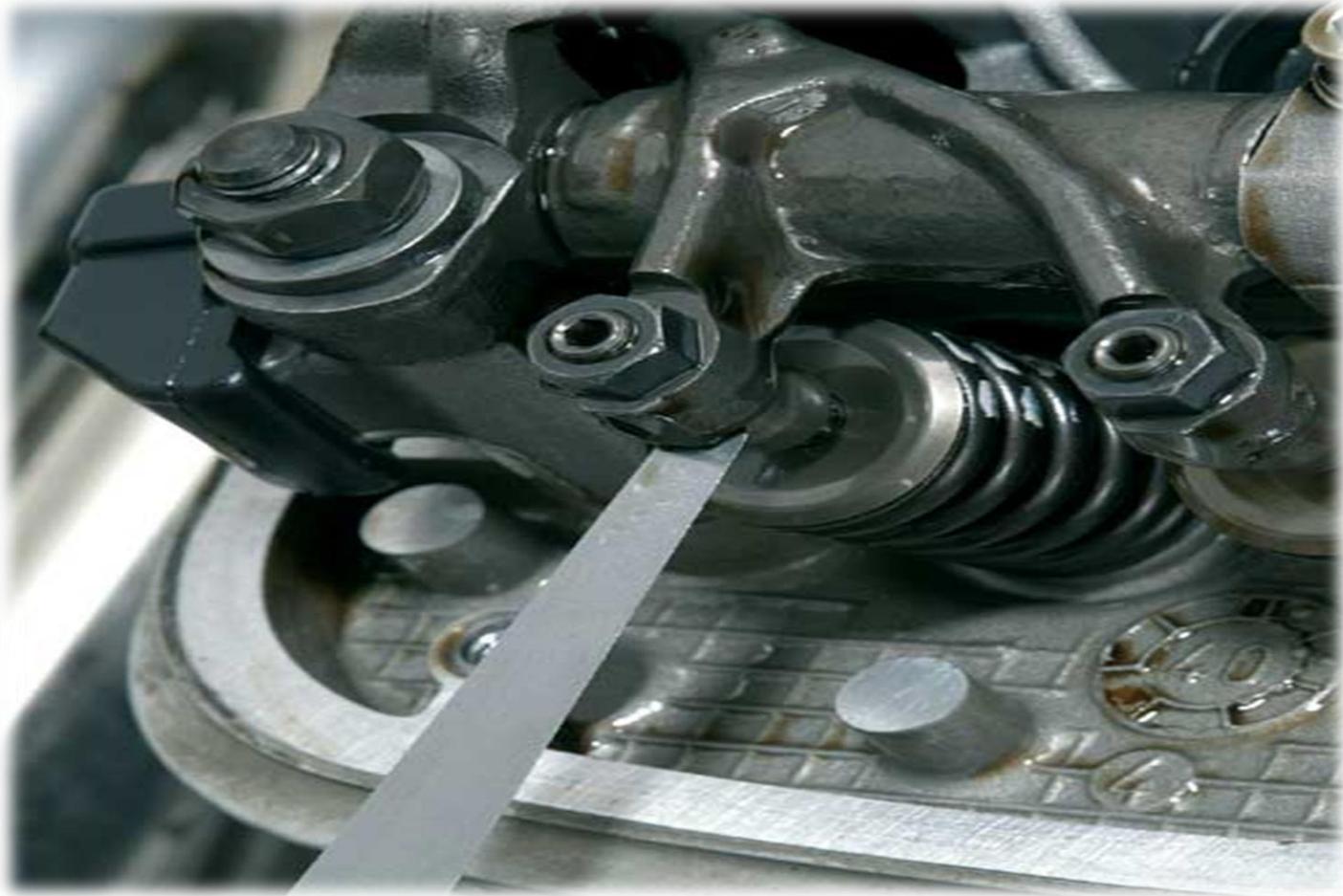
Das Auge misst mit

Visualisierung von Software-Metriken

Thomas Haug

MATHEMA Software GmbH

Motivation – Messen und Kontrollieren



Motivation – Messen von Metriken

Cyclomatic Komplexität (CC(N))

- McCabe, 1976

$$CC = e - n + 2p$$

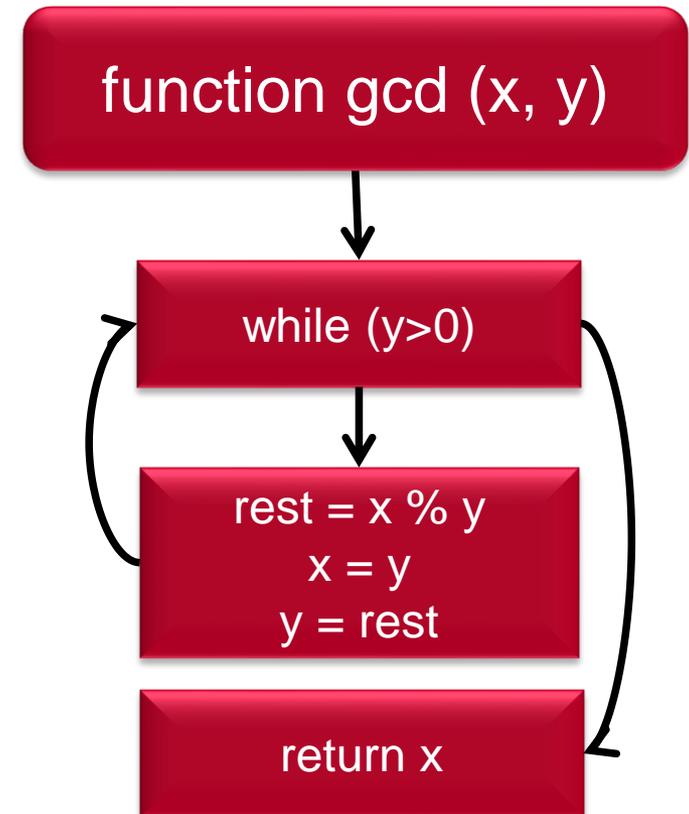
p: Anzahl der Komponenten

e: Anzahl der Kanten (edges)

n: Anzahl der Knoten (nodes)

- Beispiel

$$CC = 4 - 4 + 2 \cdot 1 = 2$$

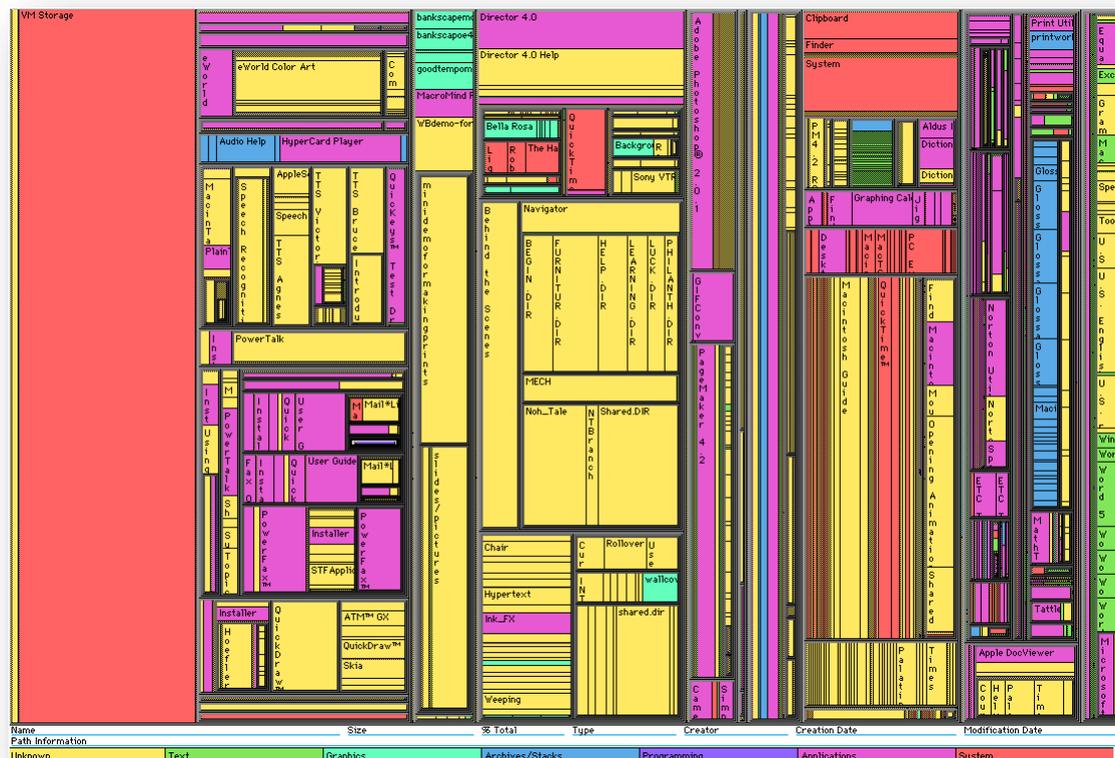


Motivation – Messen und Kontrollieren

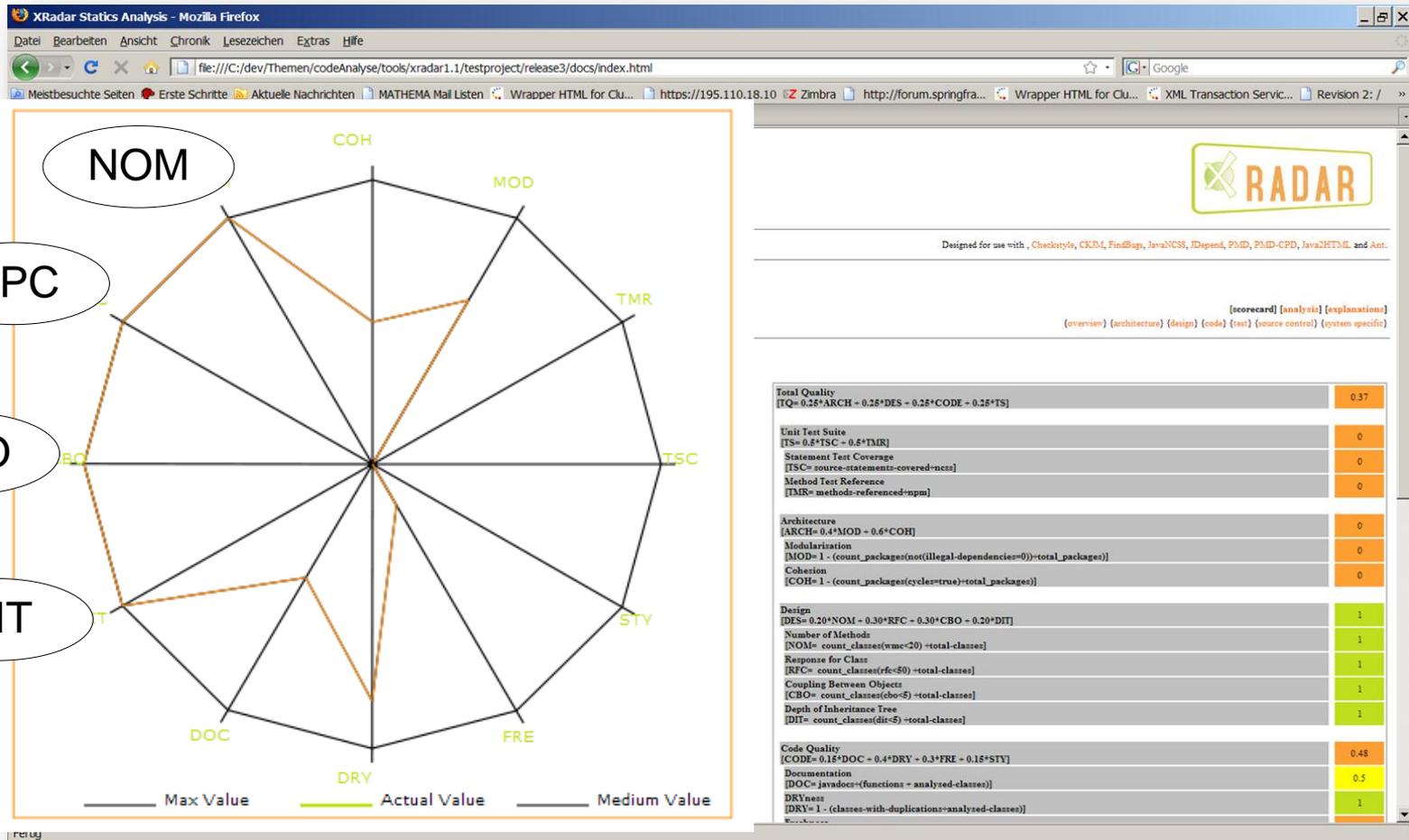


Motivation – Visualisierung

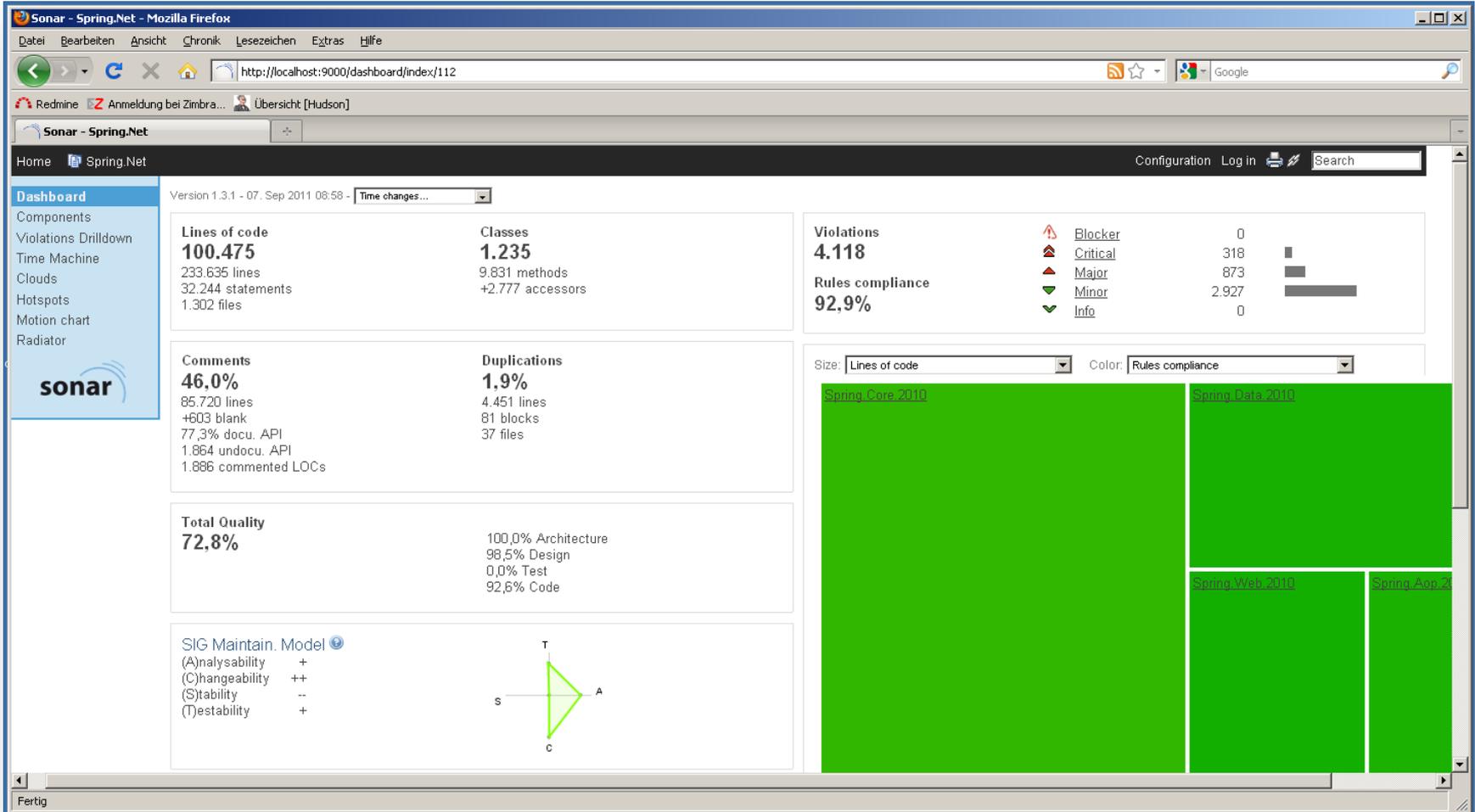
- Ben Shneiderman: “Overview first, zoom and filter, then details-on-demand” [Shn96].



Was sehen wir nicht – Kiviat Diagramme



Was sehen wir nicht – Sonar und .Net



Sonar - Spring.Net - Mozilla Firefox
 Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
 http://localhost:9000/dashboard/index/112
 Redmine Anmeldung bei Zimbra... Übersicht [Hudson]
 Sonar - Spring.Net Configuration Log in Search

Home Spring.Net

Dashboard Version 1.3.1 - 07. Sep 2011 08:58 - Time changes...

Components
 Violations Drilldown
 Time Machine
 Clouds
 Hotspots
 Motion chart
 Radiator

sonar

Lines of code
100.475
 233.635 lines
 32.244 statements
 1.302 files

Classes
1.235
 9.831 methods
 +2.777 accessors

Violations
4.118

Rules compliance
92,9%

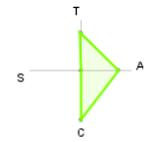
Blocker 0
 Critical 318
 Major 873
 Minor 2.927
 Info 0

Comments
46,0%
 85.720 lines
 +603 blank
 77,3% docu. API
 1.864 undocu. API
 1.886 commented LOCs

Duplications
1,9%
 4.451 lines
 81 blocks
 37 files

Total Quality
72,8%
 100,0% Architecture
 98,5% Design
 0,0% Test
 92,6% Code

SIG Maintain. Model
 (A)nalsability +
 (C)hangeability ++
 (S)tability --
 (T)estability +



Size: Lines of code Color: Rules compliance

Spring.Core.2010
 Spring.Data.2010
 Spring.Web.2010
 Spring.App.2

Fertig

Agenda

- Motivation

2D

- Overview first
- Zoom and filter
- Details-on-demand

3D

- Overview first
- Zoom and filter
- Details-on-demand

- Zusammenfassung

Agenda

- Motivation

2D

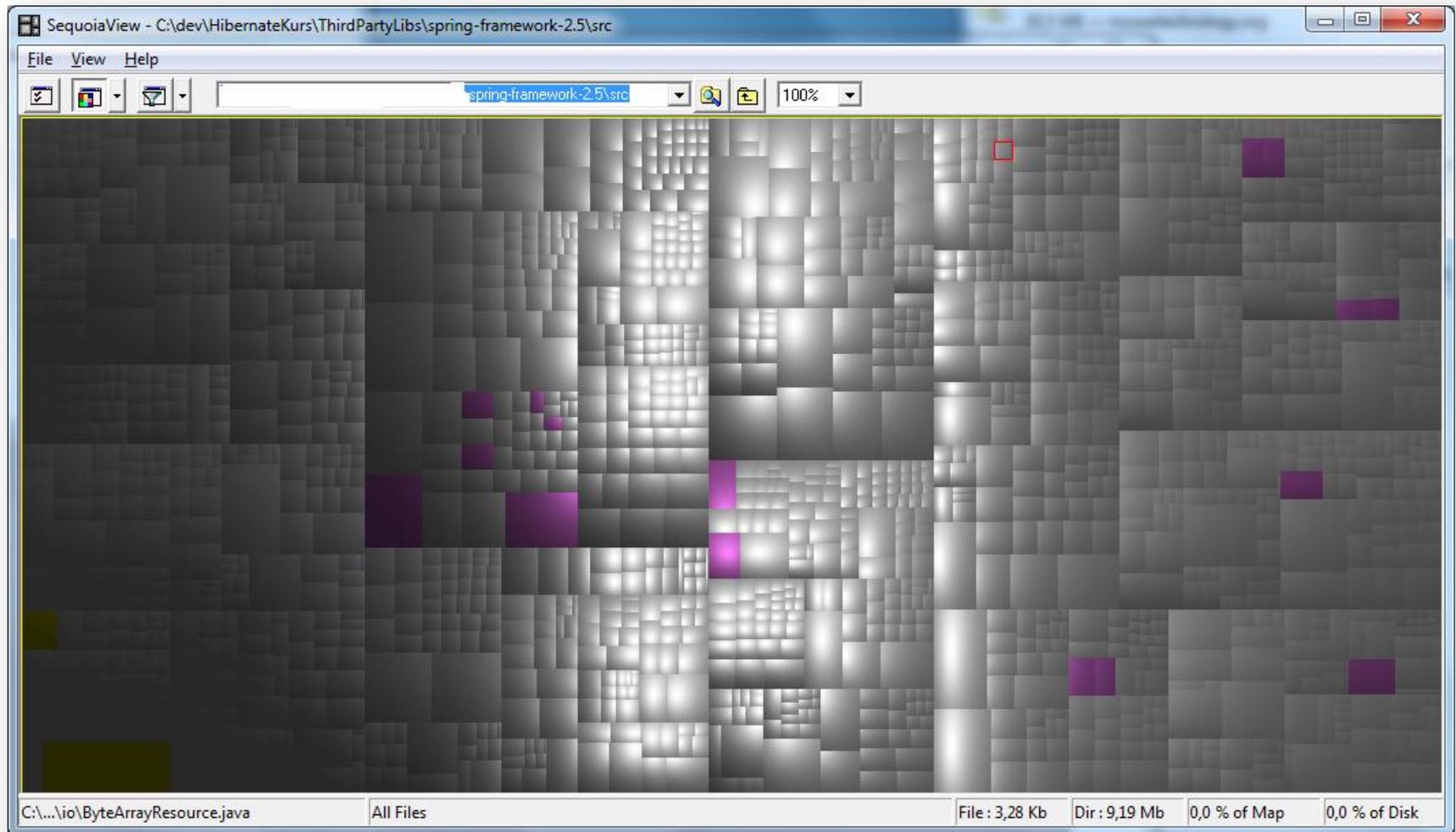
- Overview first
- Zoom and filter
- Details-on-demand

3D

- Overview first
- Zoom and filter
- Details-on-demand

- Zusammenfassung

TreeMap – SequoiaView Spring 2.5



TreeMap – NDepend Beispiel

Spring.Net1.3.1 (analysis done 13 hours ago, Fri 20 May 20:26 most recent)

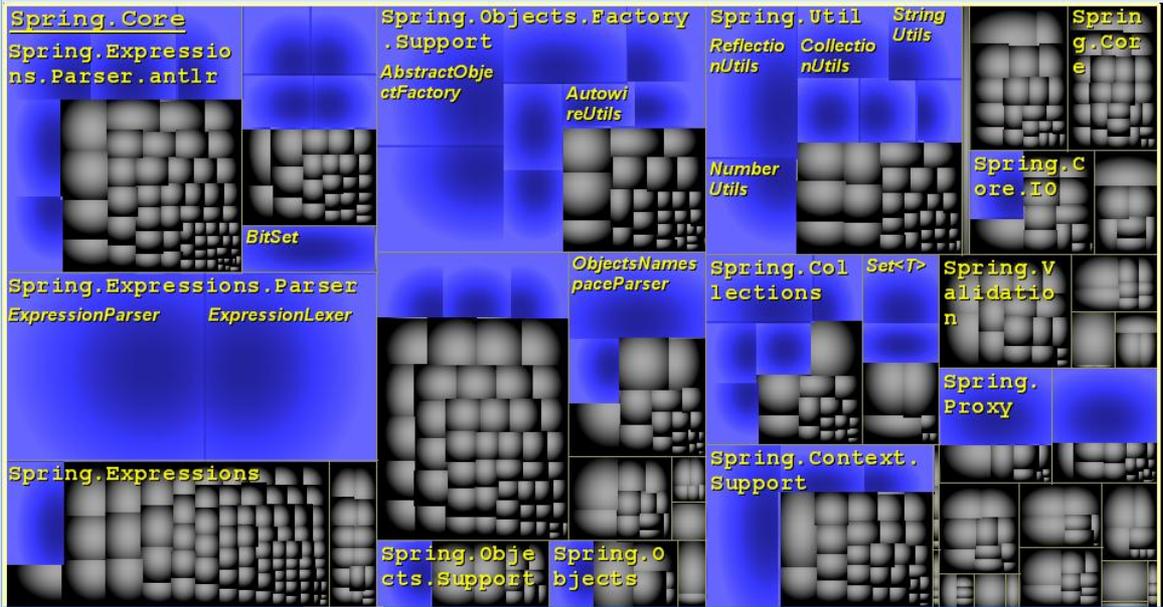
File View Analysis Tools Compare Coverage Help

CQL Query Edit* 48 Items

SELECT TYPES WHERE ILCCyclomaticComplexity > 50
ORDER BY ILCCyclomaticComplexity DESC

Metrics

Level: Type Metric: IL Cyclomatic Complexity (ILCC): 1 unit = 38,1 pi...



Spring.Core
Spring.Expressions.Parser antlr
Spring.Expressions.Parser ExpressionParser
ExpressionLexer
BitSet
Spring.Expressions
Spring.Objects.Factory.Support
AbstractObjectFactory
AutowireUtils
ObjectsNamespaceParser
Spring.Objects.Support
ReflectionUtils
CollectionUtils
StringUtils
NumberUtils
Spring.Collections
Set
Spring.Collections
Spring.Validation
Spring.Proxy
Spring.Context.Support
Spring.Objects.Support
Spring.Objects

types	IL Cyclomatic Complexity (ILCC)
48 types matched	
ExpressionParser	684
ExpressionLexer	585
AbstractObjectFactory	331
ReflectionUtils	254
AbstractAutowireCapableObjectFactory	245
AbstractApplicationContext	215
ObjectsNamespaceParser	212
DefaultListableObjectFactory	175
NumberUtils	159
AbstractProxyTypeBuilder	156
CharScanner	155
BaseAST	150
DynamicReflectionManager	140
PriorityQueue	131
CollectionUtils	127

Dependency Graph | Dependency Matrix | Metrics | Project Properties

CQL Query Explorer

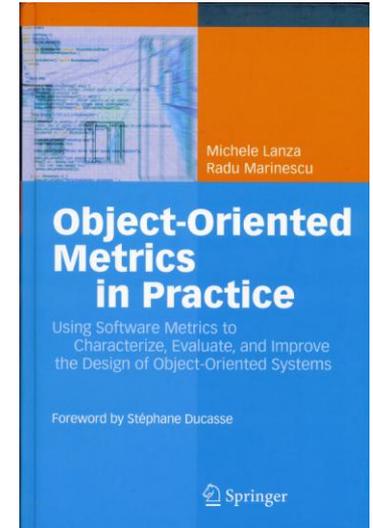
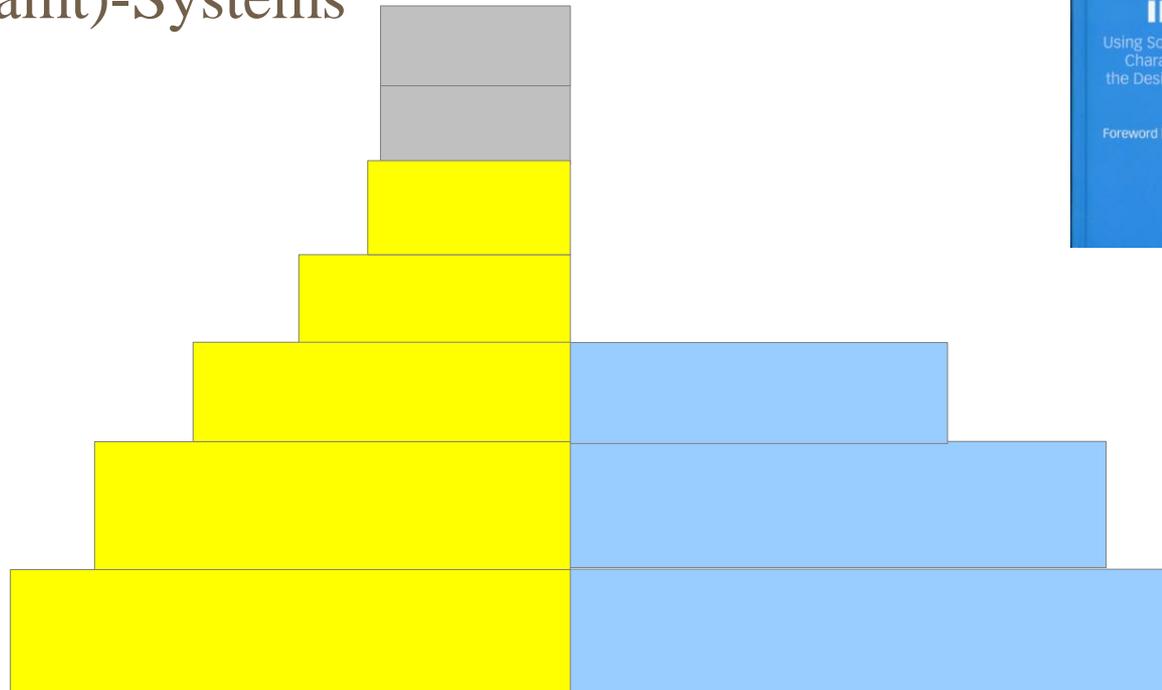
Create Group | Delete Group

Create Query | Delete Query

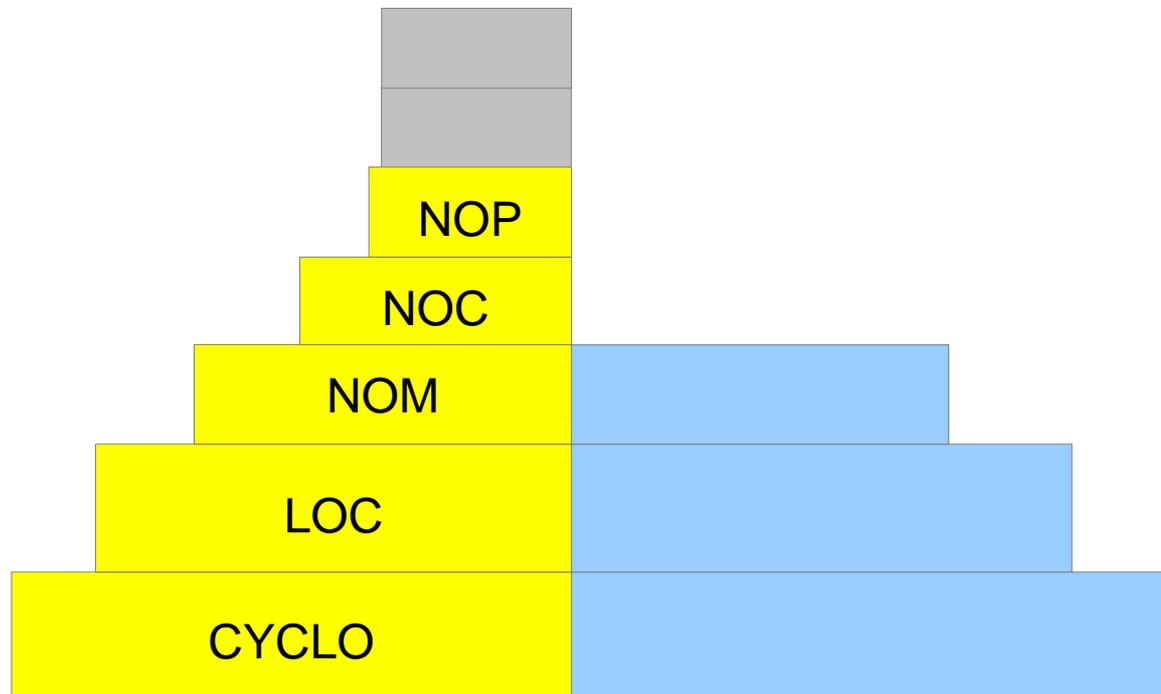
Active	#Items	CQL™ Query/Constraint
<input checked="" type="checkbox"/>	1	Methods too complex - critical (ILCyclomaticComplexity)
<input checked="" type="checkbox"/>	0	Methods with too many parameters - critical (NbParam)

Pyramidal Overview

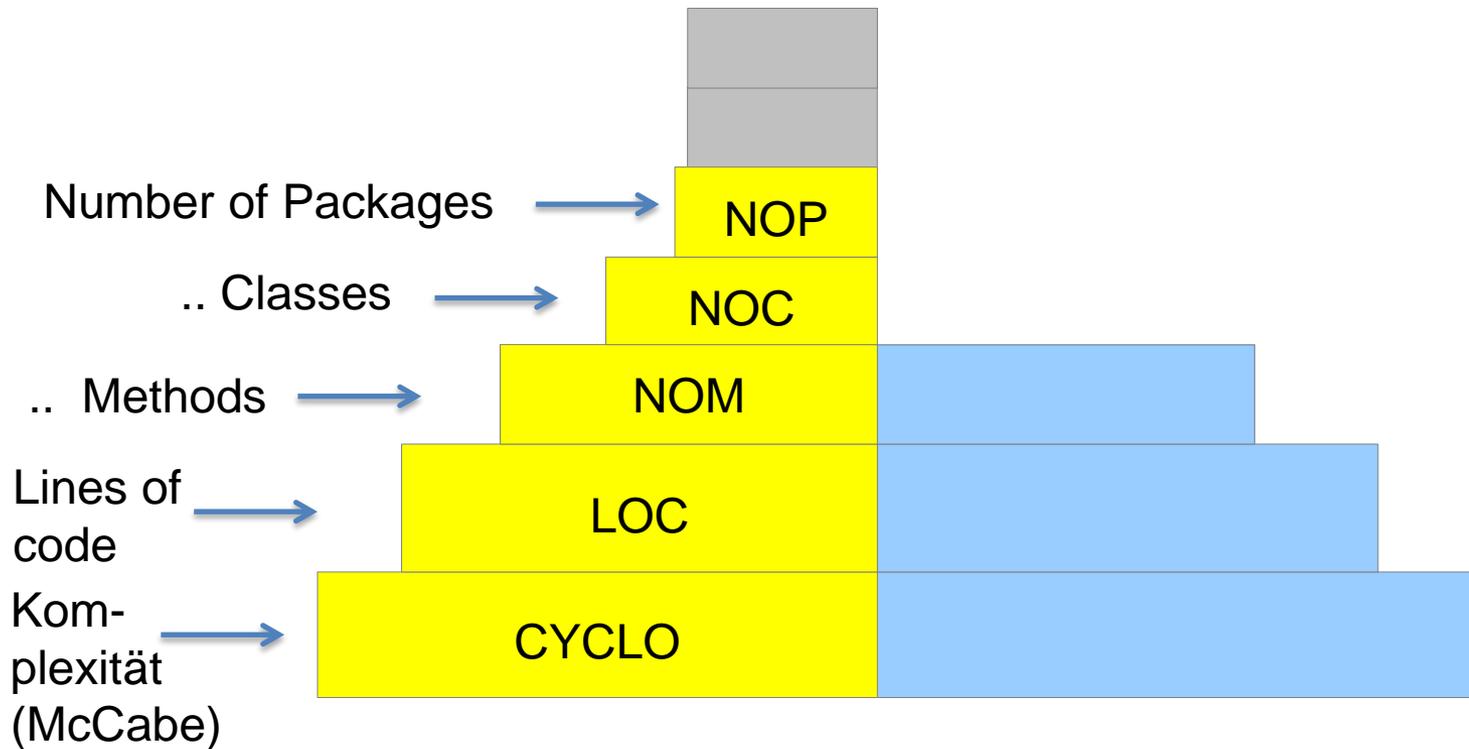
- Lanza und Marinescu (2006)
- „Referenzmodell“ für Java und C++
- Bewertet die Strukturierung des (Gesamt)-Systems



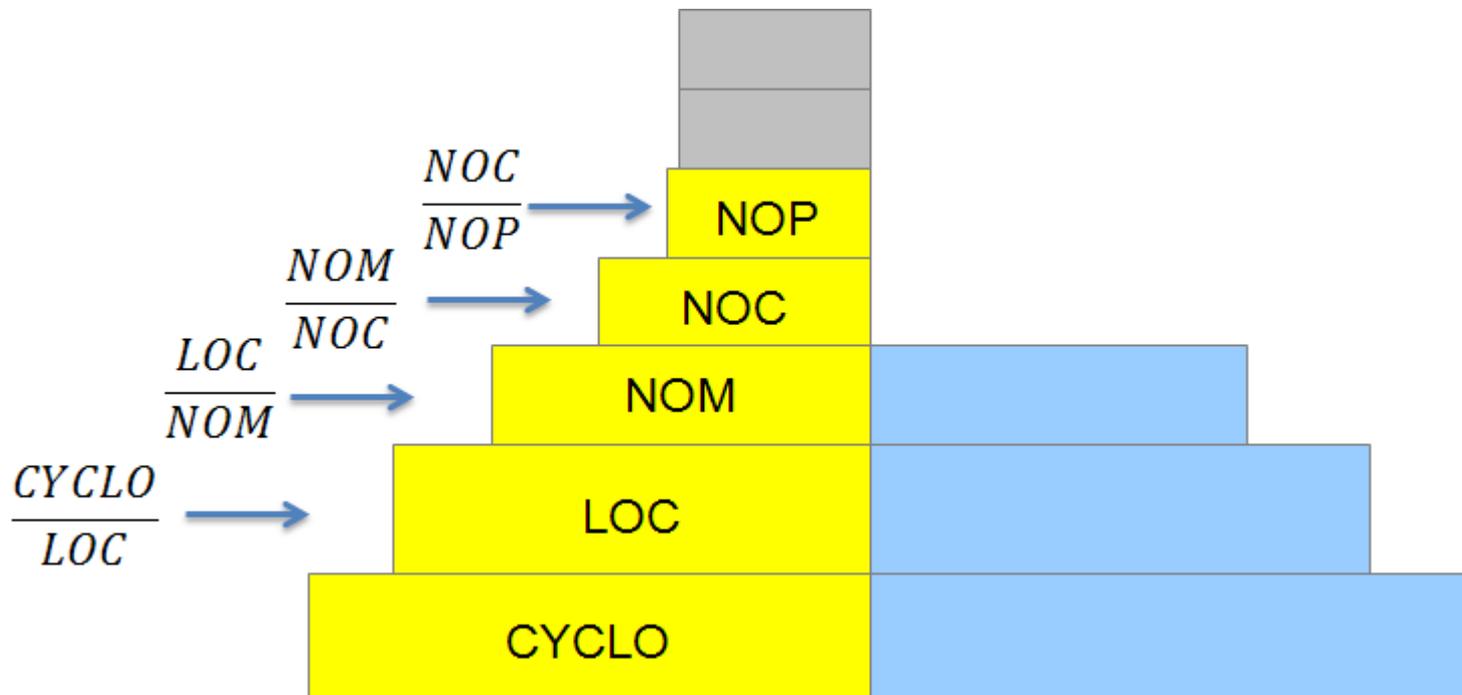
Pyramidal Overview – Größe und Komplexität



Pyramidal Overview – Größe und Komplexität



Pyramidal Overview – Größe und Komplexität

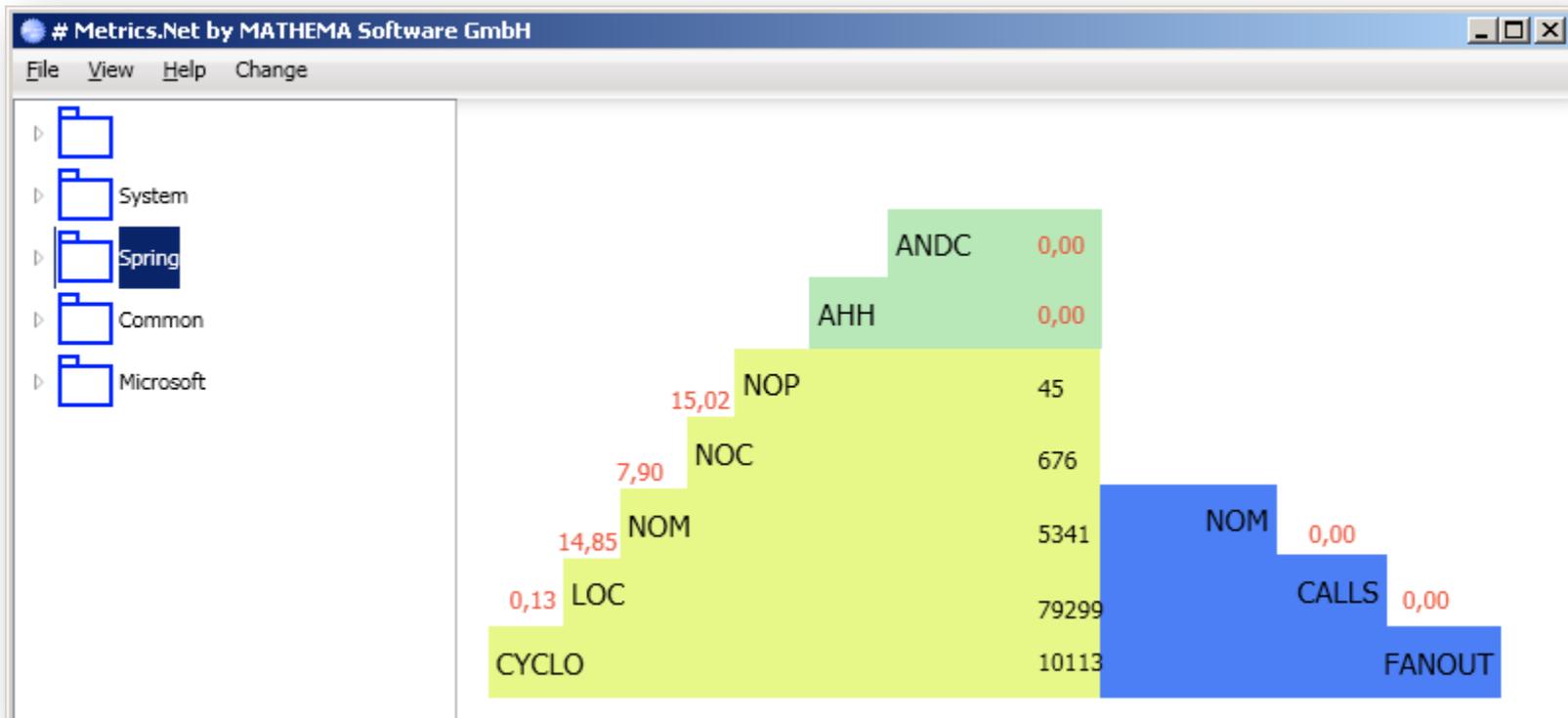


Pyramidal Overview – Größe und Komplexität

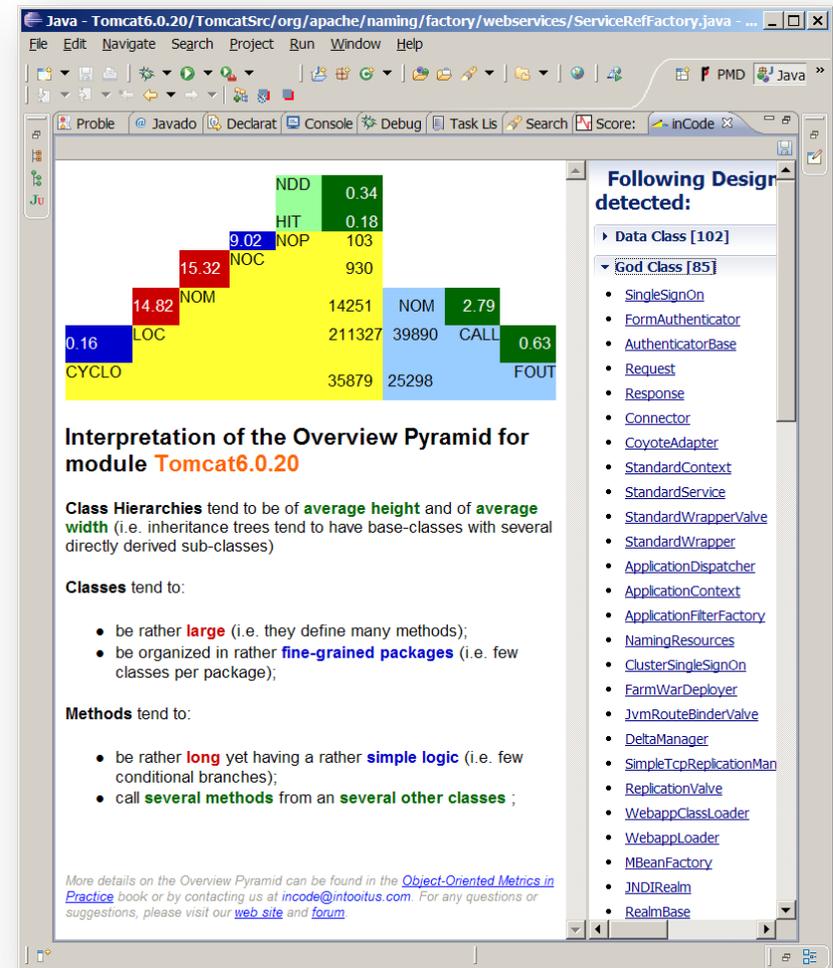
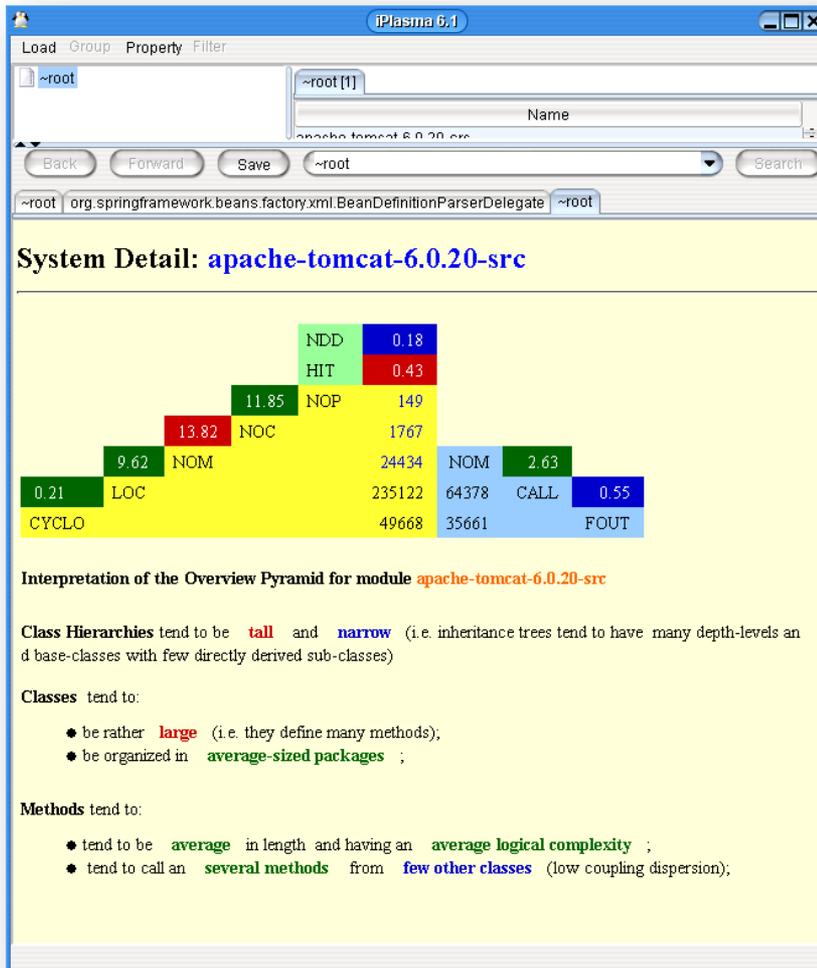
Java Projekte

Metrik	Niedrig	Durchschnitt	Hoch
$\frac{NOC}{NOP}$	6	17	26
$\frac{NOM}{NOC}$	4	7	10
$\frac{LOC}{NOM}$	7	10	13
$\frac{CYCLO}{LOC}$	0,16	0,20	0,24

Pyramidal Overview – Beispiel Spring.Net



Pyramidal Overview – Beispiel Tomcat



Pyramidal Overview – Beispiel Tomcat

iPlasma 6.1

Load Group Property Filter

~root

~root [1]

Back Forward Save ~root

~root org.springframework.beans.factory.xml.BeanDefinition

System Detail: apache-tomcat-6.0

	NDD	0.18
	HIT	0.43
	11.85	NOP 149
	13.82	NOC 1767
	9.62	NOM 24434
	0.21	LOC 235122
		CYCLO 49668

Interpretation of the Overview Pyramid for module **ap:**

Class Hierarchies tend to be **tall** and **narrow** (i.e. d base-classes with few directly derived sub-classes)

Classes tend to:

- ◆ be rather **large** (i.e. they define many methods);
- ◆ be organized in **average-sized packages** ;

Methods tend to:

- ◆ tend to be **average** in length and having an a
- ◆ tend to call an **several methods** from **few other classes** (low coupling dispersion);



Java - Tomcat6.0.20/TomcatSrc/org/apache/naming/factory/webservices/ServiceRefFactory.java - ...

File Edit Navigate Search Project Run Window Help

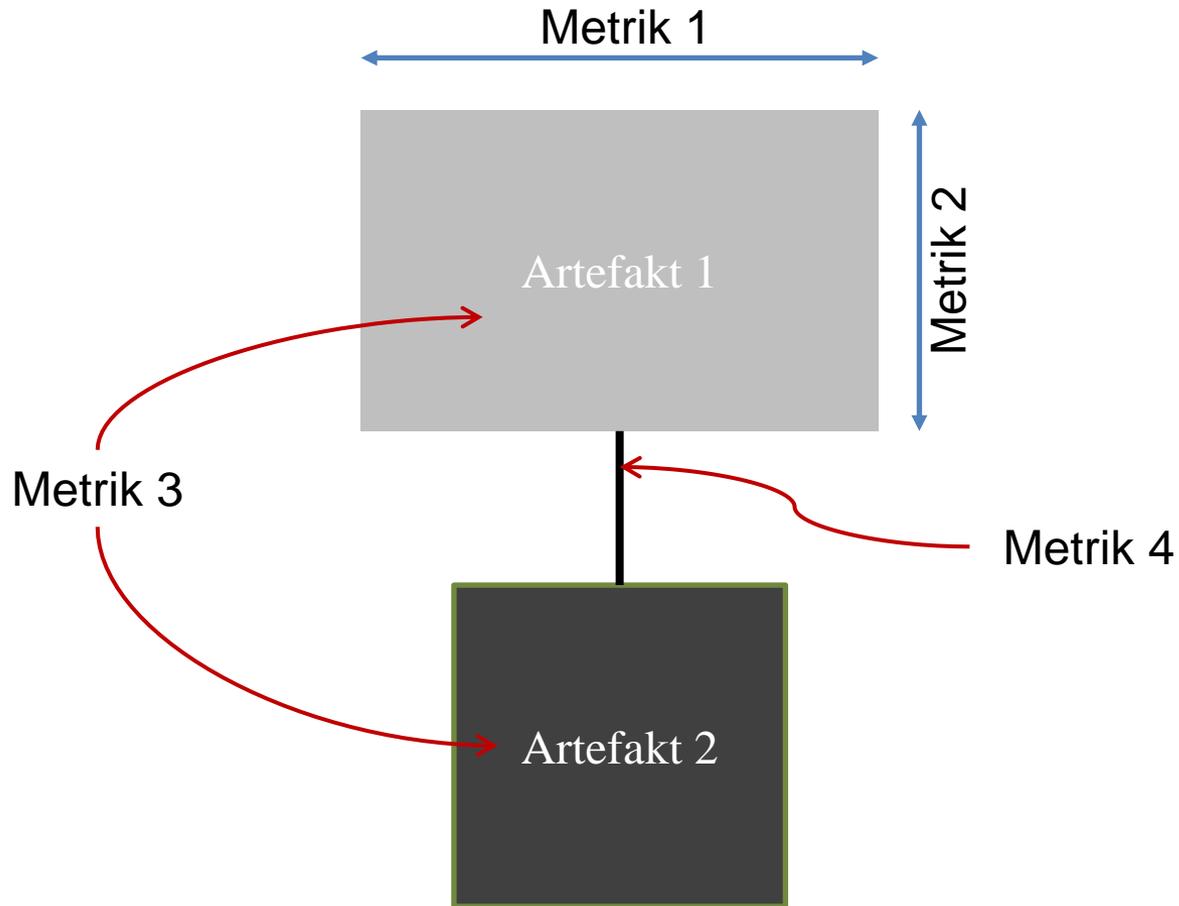
Task Lis Search Score: inCode

Following Design detected:

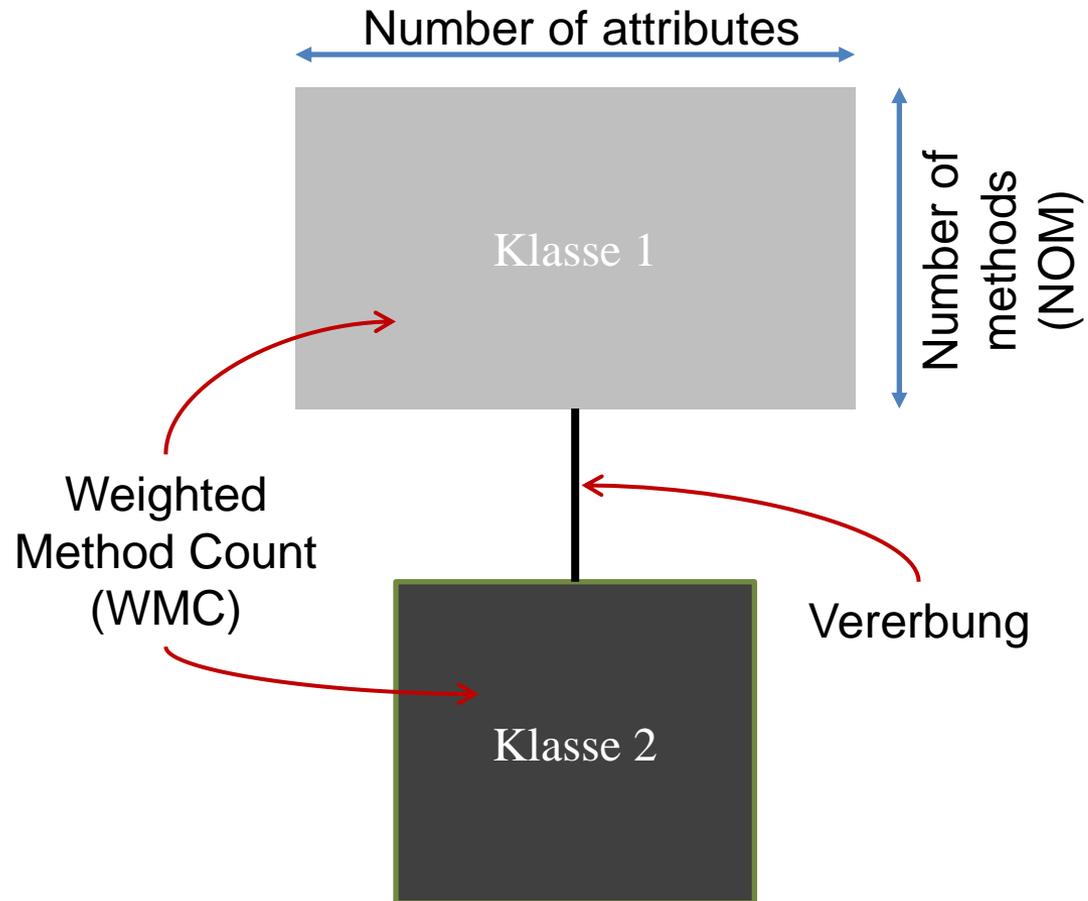
- ▶ **Data Class [102]**
- ▶ **God Class [85]**
 - SingleSignOn
 - FormAuthenticator
 - AuthenticatorBase
 - Request
 - Response
 - Connector
 - CoyoteAdapter
 - StandardContext
 - StandardService
 - StandardWrapperValve
 - StandardWrapper
 - ApplicationDispatcher
 - ApplicationContext
 - ApplicationFilterFactory
 - NamingResources
 - ClusterSingleSignOn
 - FarmWarDeployer
 - JvmRouteBinderValve
 - DeltaManager
 - SimpleTopReplicationMan
 - ReplicationValve
 - WebappClassLoader
 - WebappLoader
 - MBeanFactory
 - JNDIRealm
 - RealmBase

More details on the Overview Pyramid can be found in the [Object-Oriented Metrics in Practice](#) book or by contacting us at incode@intoitus.com. For any questions or suggestions, please visit our [web site](#) and [forum](#).

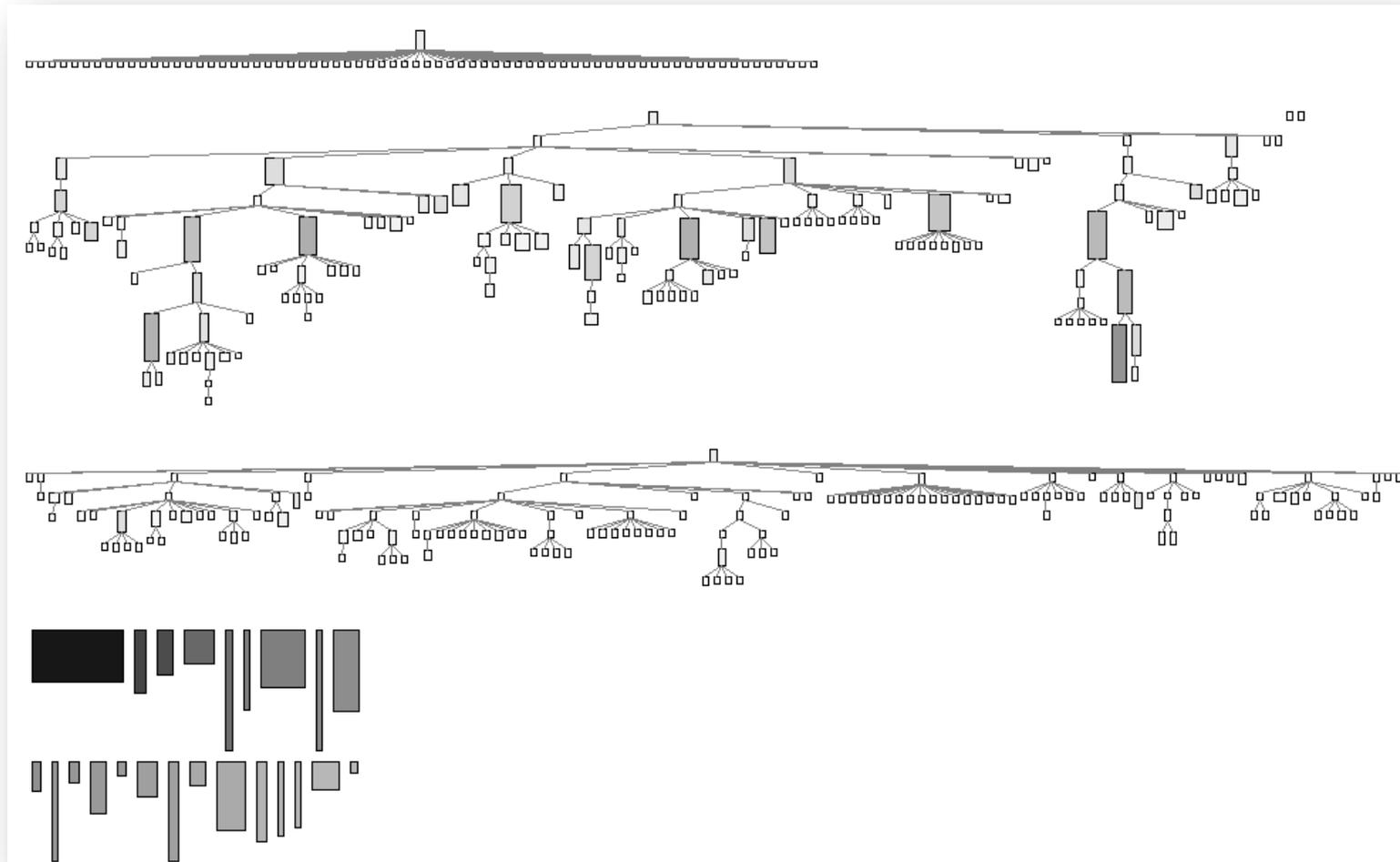
Polymetric View



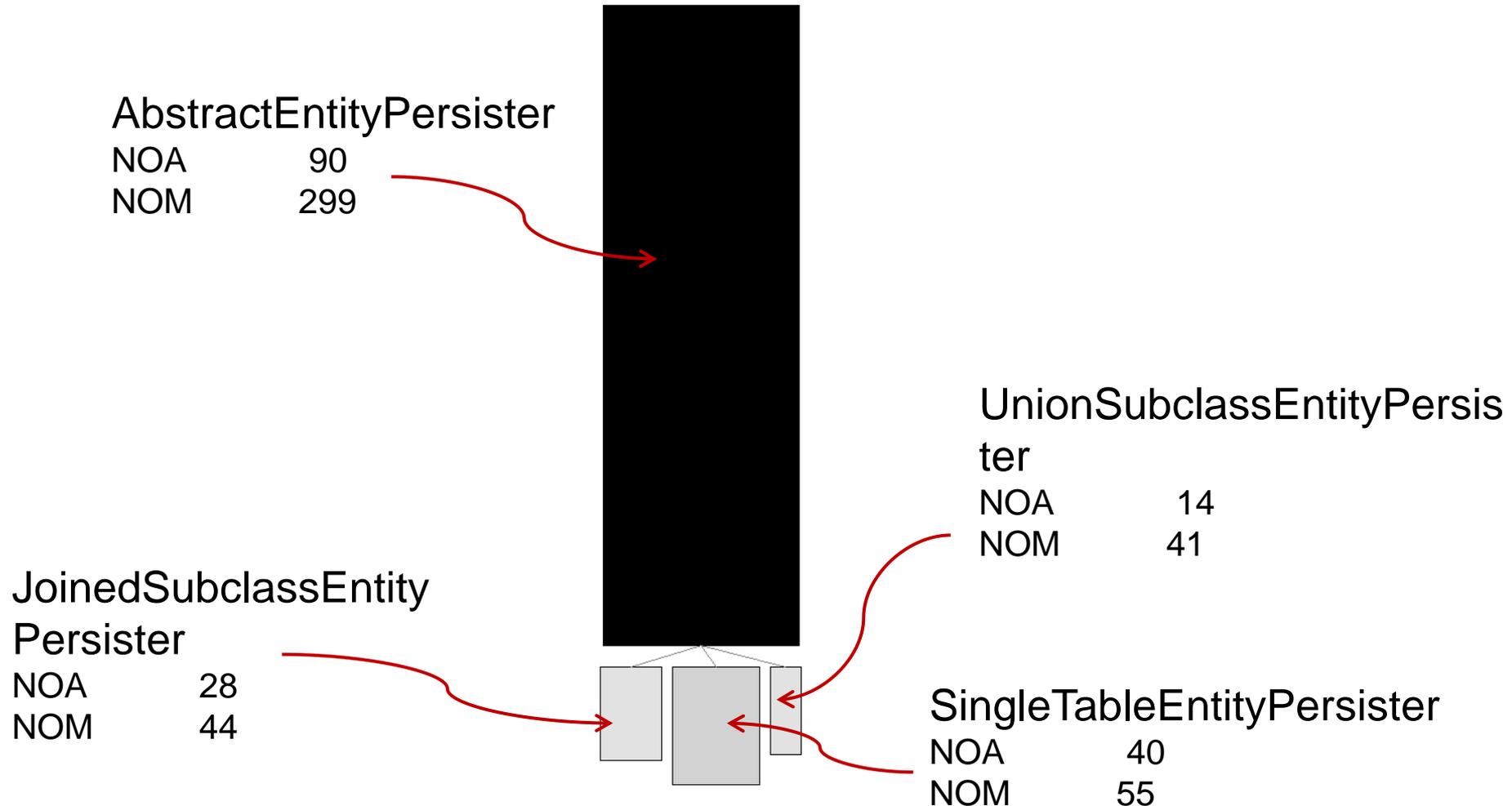
Polymetric View – System Complexity View



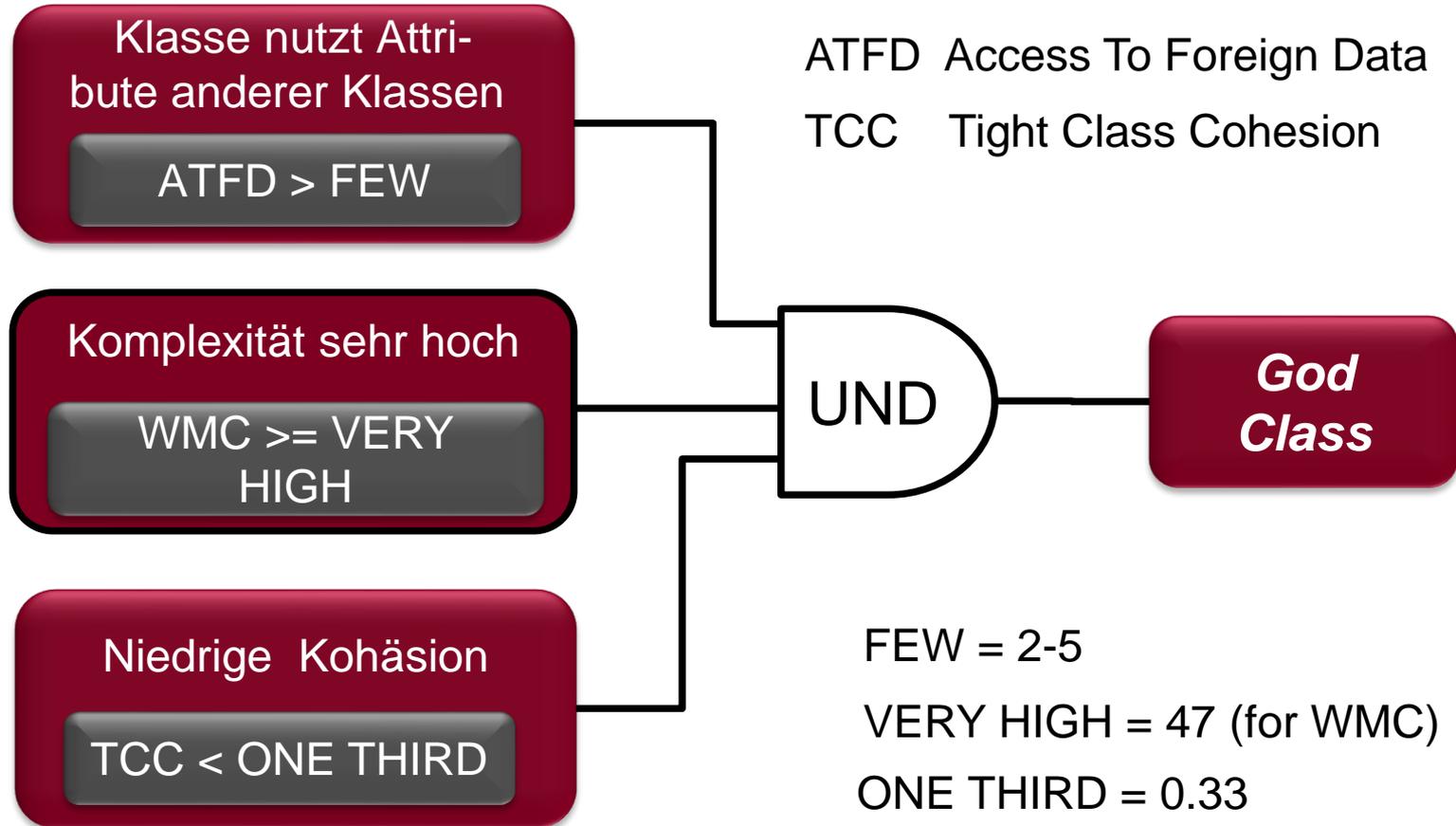
Polymetric View – System Complexity View



Polymetric View – System Complexity View



Polymetric View – Disharmonies (Lanza)



Polymetric View – Disharmonies Darstellung

identity
disharmony

- Feature Envy (Fowler)
- Data Class (Fowler)
- God class (Riel / Fowler)
- Brain Method
- Brain Class
- Significant Duplication (Fowler)

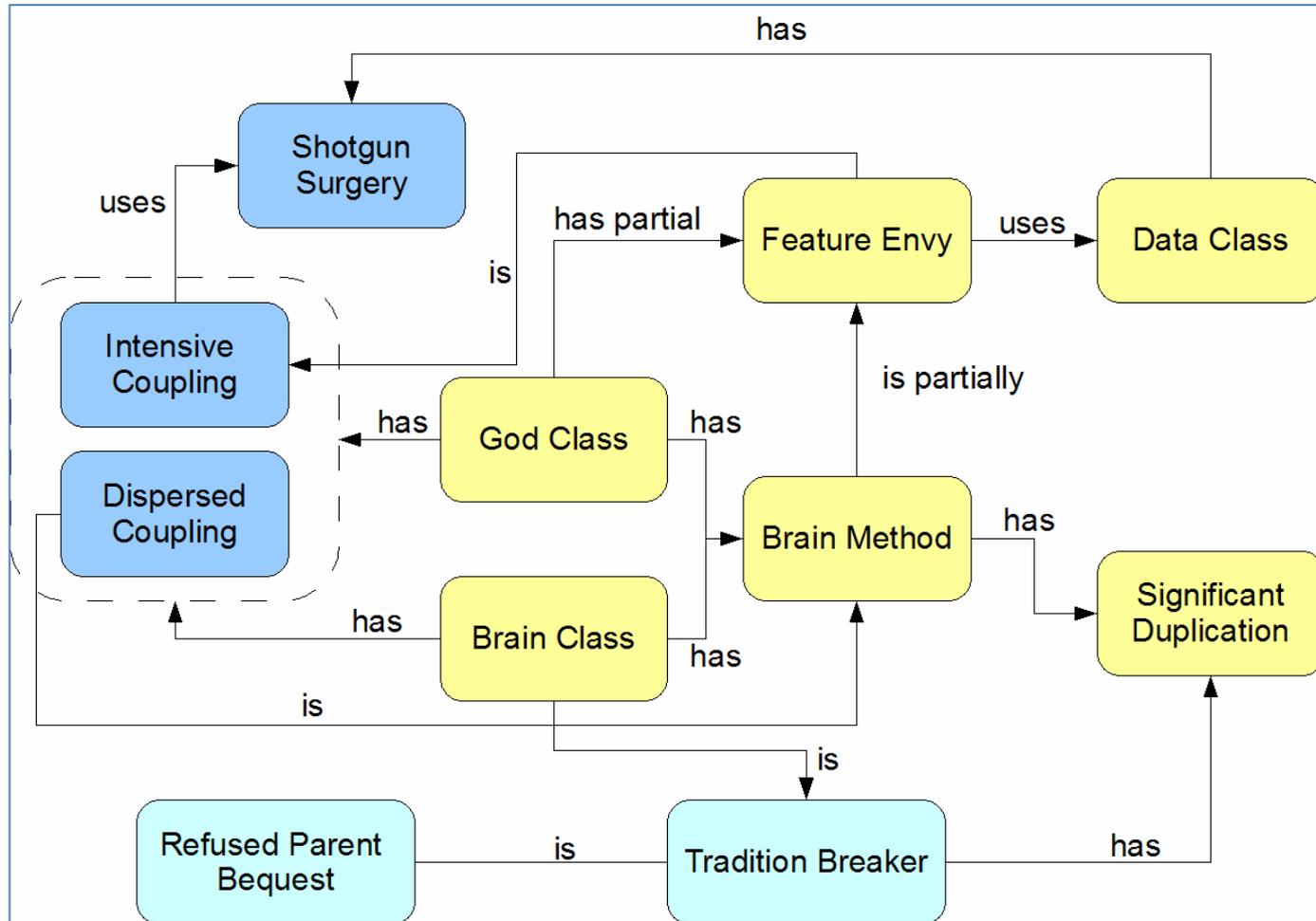
collaboration
disharmony

- Shotgun Surgery (Fowler)
- Intensive Coupling (Fowler)
- Dispersed Coupling

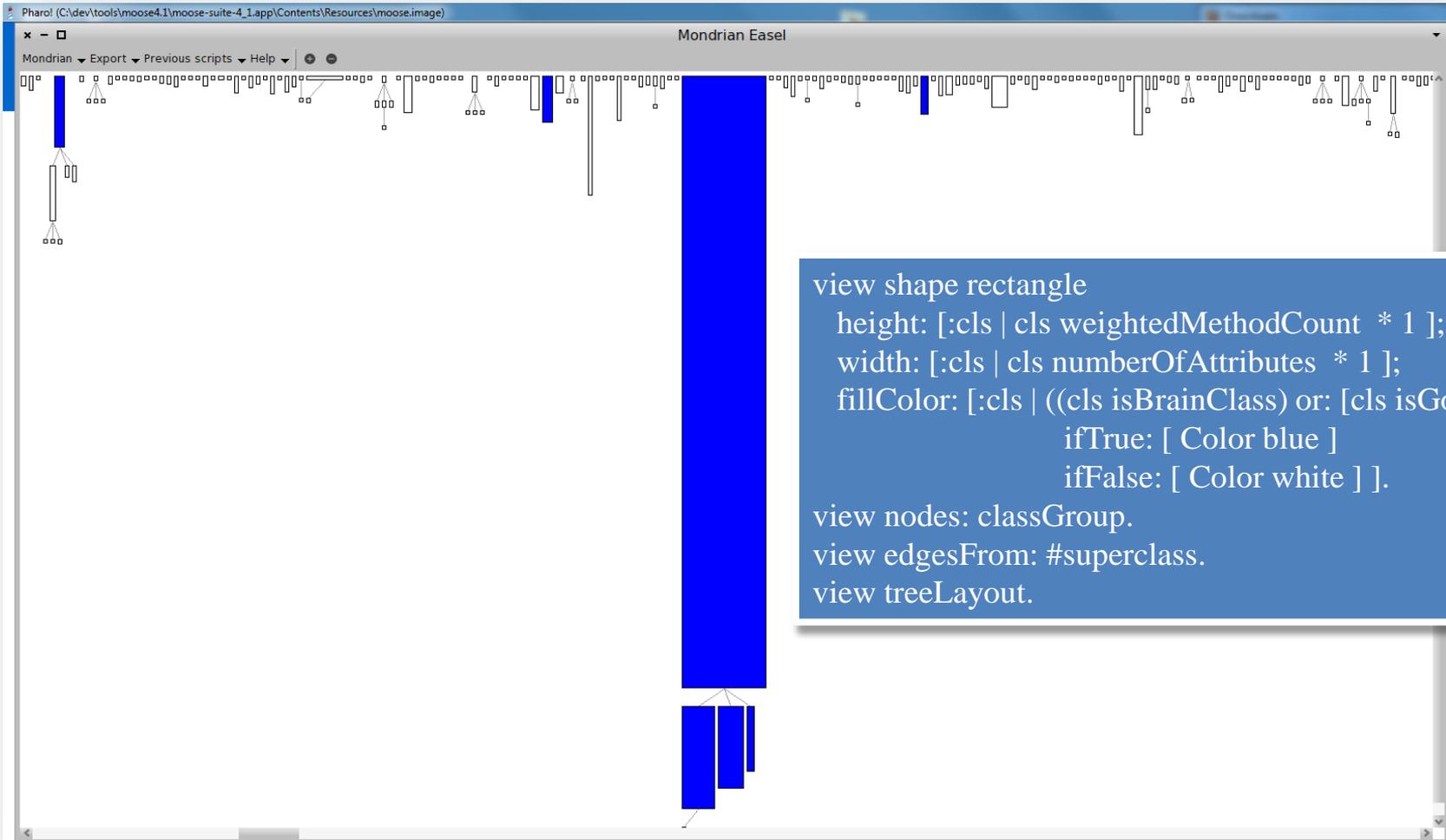
classification
disharmony

- Refused Parent Bequest (Fowler)
- Tradition Breaker

Polymetric View – Disharmony Beispiel



Polymetric View – Eigene polymetrische Views



The screenshot shows the Mondrian Easel application window. The title bar reads "Mondrian Easel". The menu bar includes "Mondrian", "Export", "Previous scripts", and "Help". The main area displays a class hierarchy diagram with nodes and edges. A large blue rectangle highlights a specific node in the hierarchy, which is a class group. Below this node, three smaller blue rectangles represent its subclasses.

```

view shape rectangle
  height: [:cls | cls weightedMethodCount * 1 ];
  width: [:cls | cls numberOfAttributes * 1 ];
  fillColor: [:cls | ((cls isBrainClass) or: [cls isGodClass] )
    ifTrue: [ Color blue ]
    ifFalse: [ Color white ] ].

view nodes: classGroup.
view edgesFrom: #superclass.
view treeLayout.
  
```

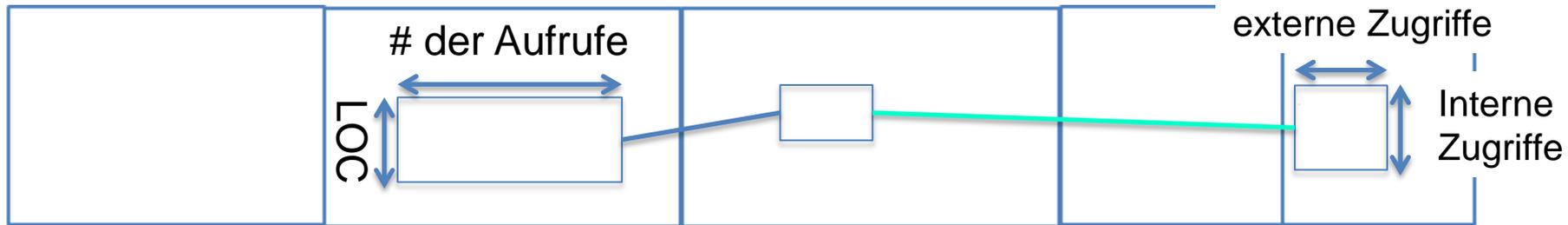
Class Blueprint

- Ermöglicht die Analyse des Klassendesigns
- Versucht den Programmablauf in einer Klasse zu visualisieren

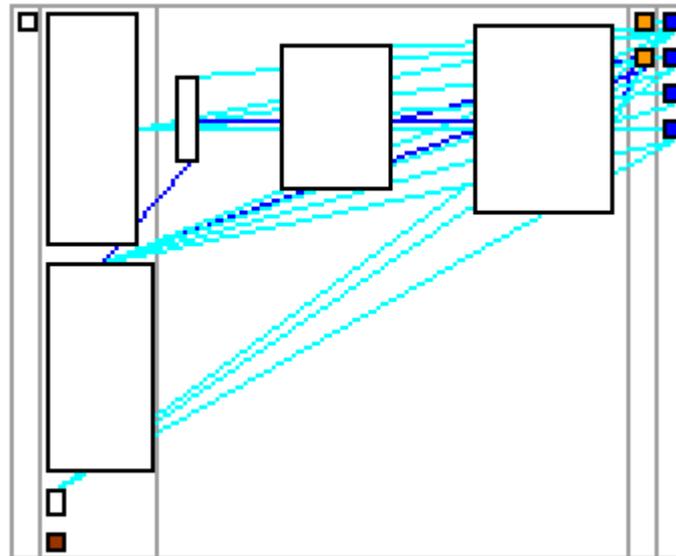
Initialisierungsmethoden	Externe Schnittstellenmethoden	Interne Implementierungsmethoden	Zugriffsmethoden	Attribute
--------------------------	--------------------------------	----------------------------------	------------------	-----------

- Öffentlichkeit nimmt von links nach rechts ab.

Class Blueprint

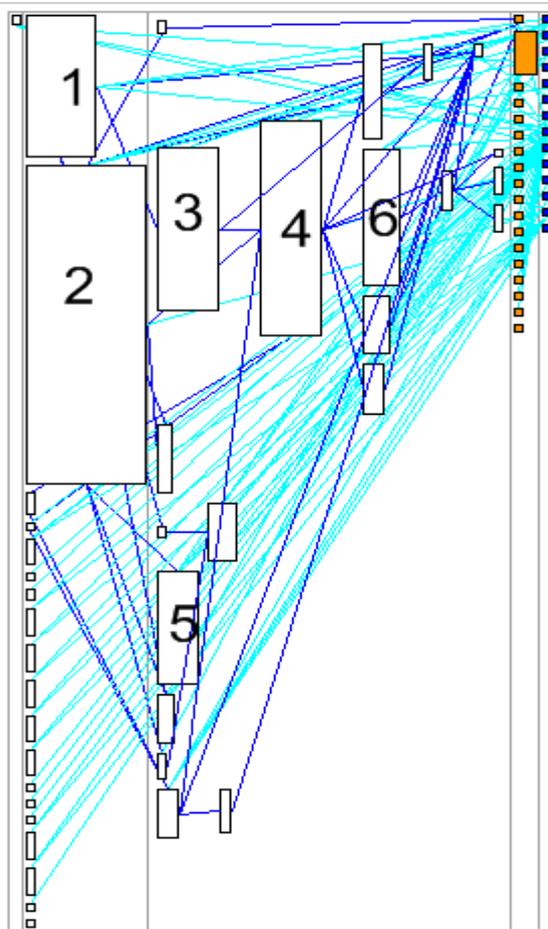


- Beispiel



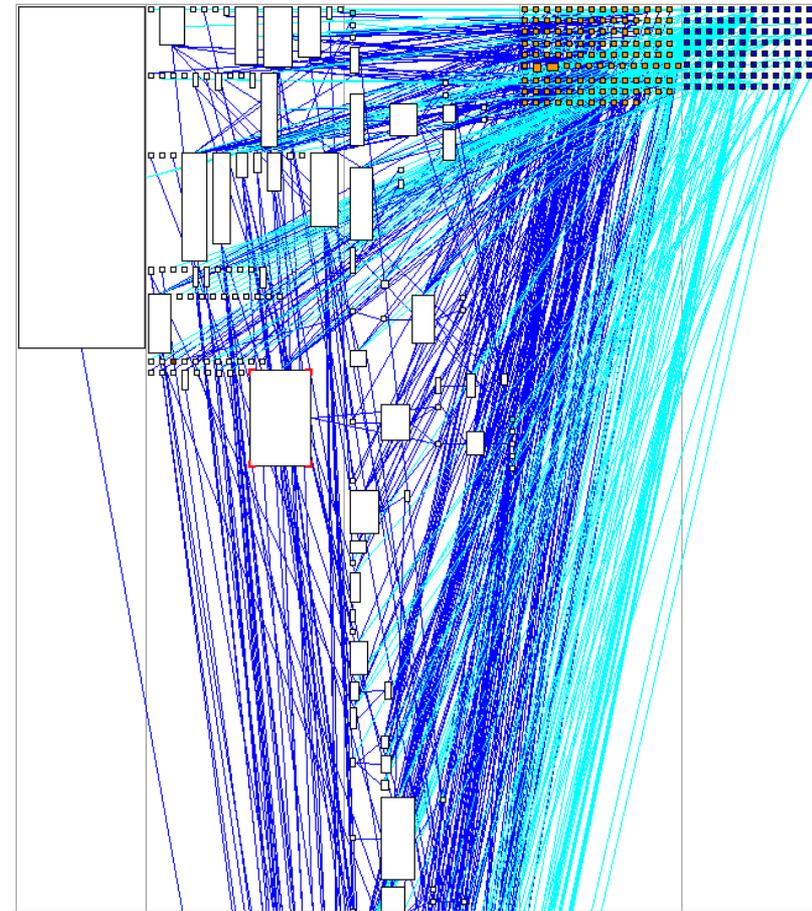
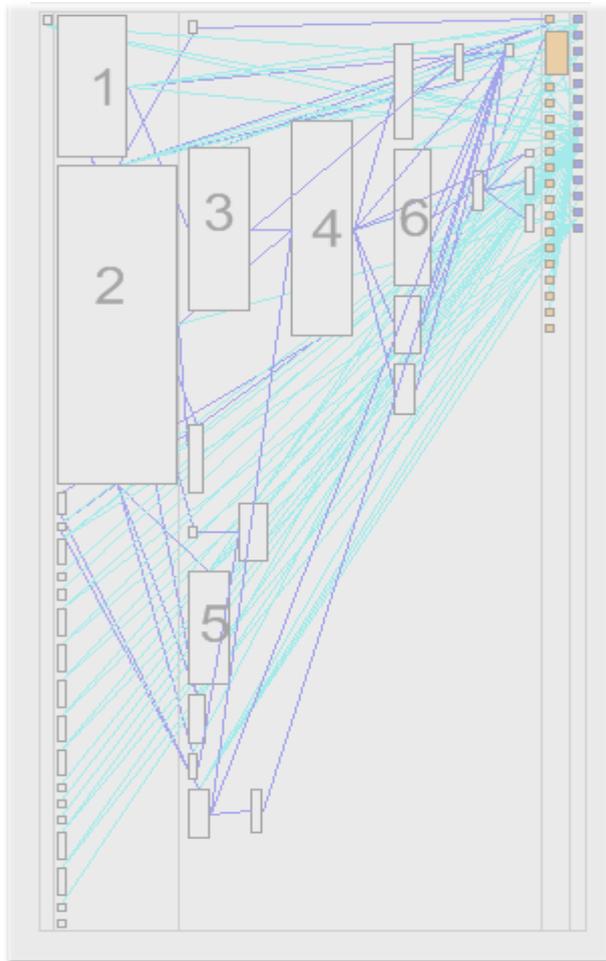
Class Blueprint – Hibernate 3.5.6

EjbConfiguration (Brain Methoden)



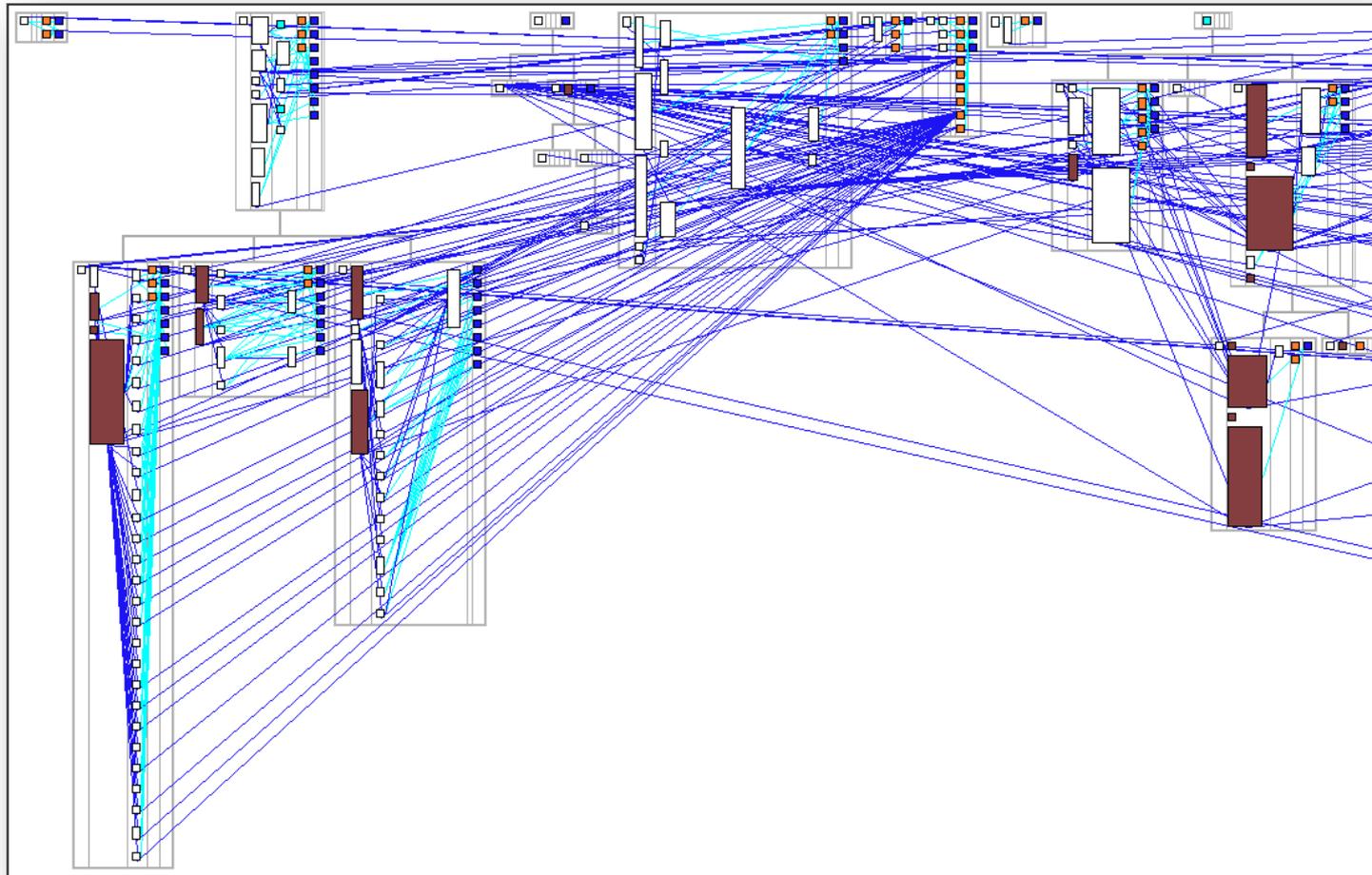
- Methoden 1-6 sind sehr groß
- Markierte Methoden greifen auf viele Attribute der Klasse zu
- Markierte Methoden sind *Brain Methods* :
 - Line of Code (LOC) > 65
 - zyklomatische Komplexität > 3
 - Verschachtelungstiefe > 3
 - Verwendet mehr als 7-8 Attribute

Class Blueprint – Hibernate 3.5.6



...

Class Blueprint – Auf Package Ebene



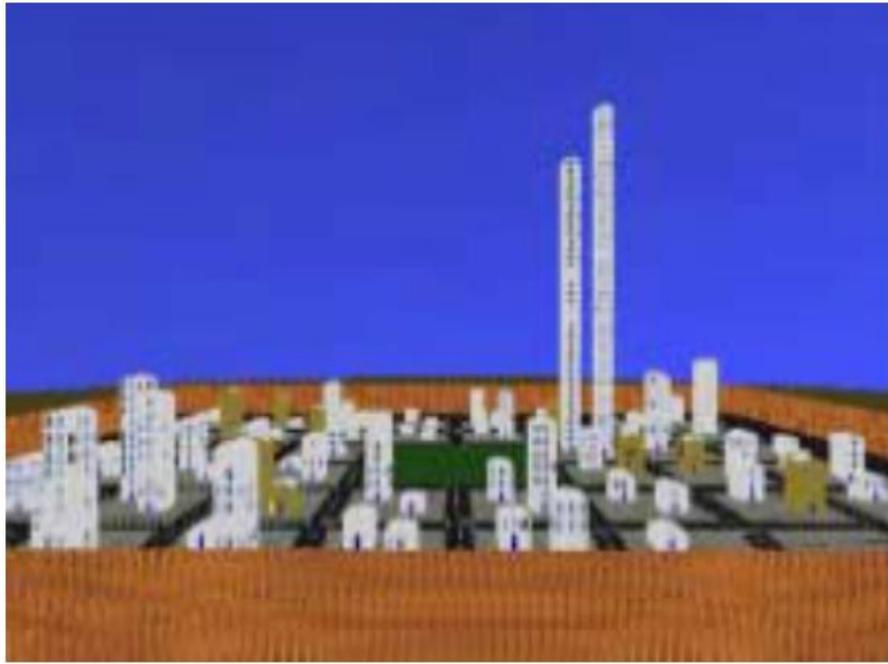
Agenda

- Motivation
- 2D
- Overview first
- Zoom and filter
- Details-on-demand

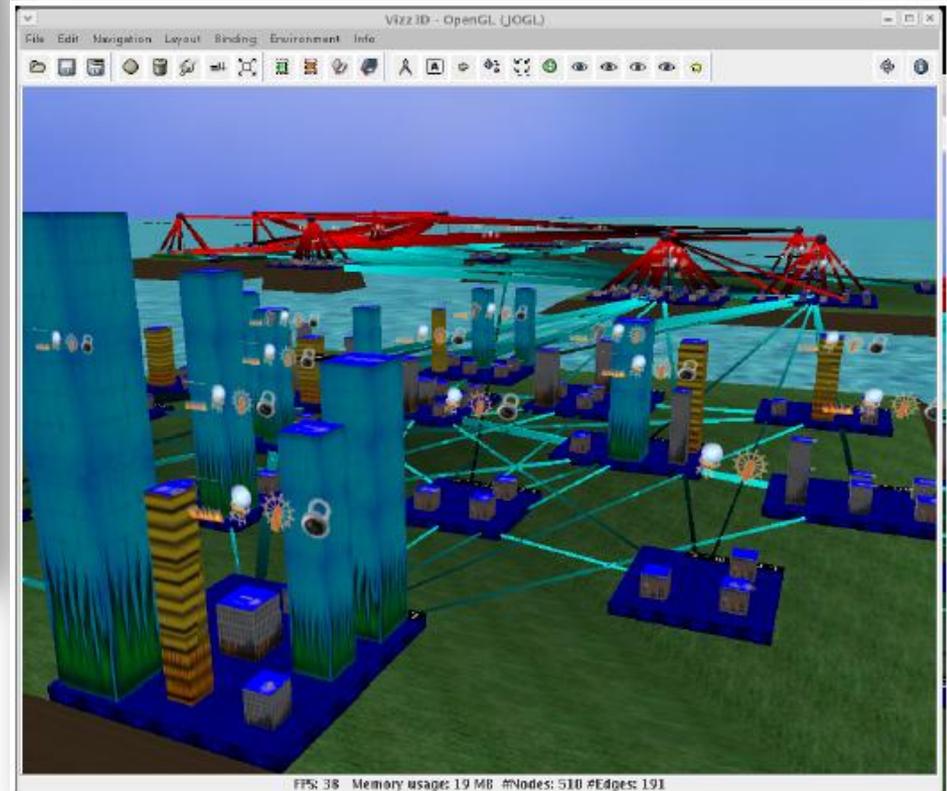
3D

- Overview first
- Zoom and filter
- Details-on-demand
- Zusammenfassung

3D – Städte Metapher

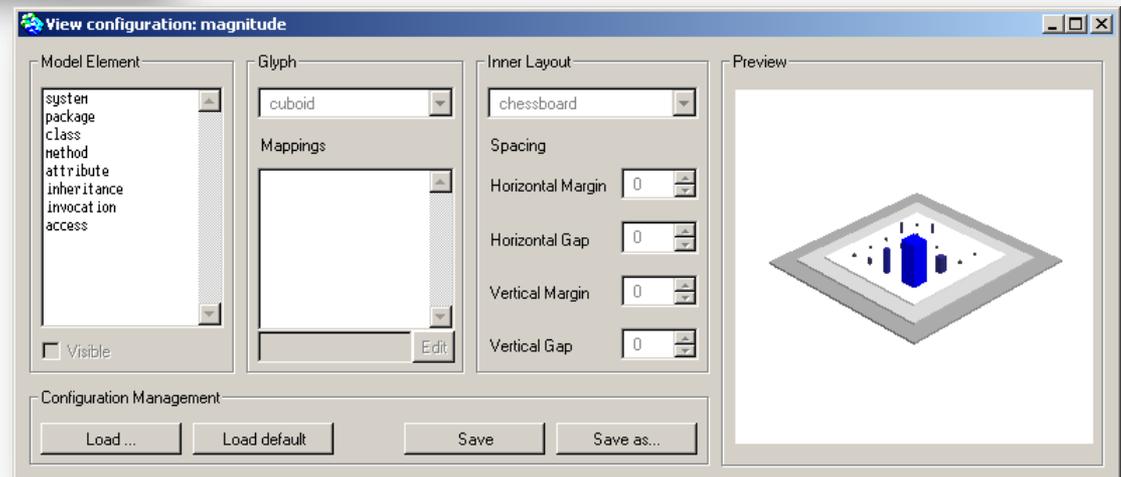
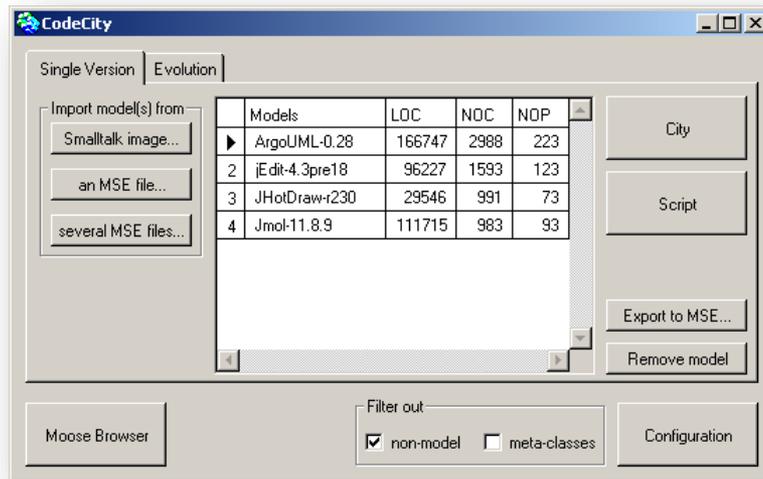


Software Worlds (2000)

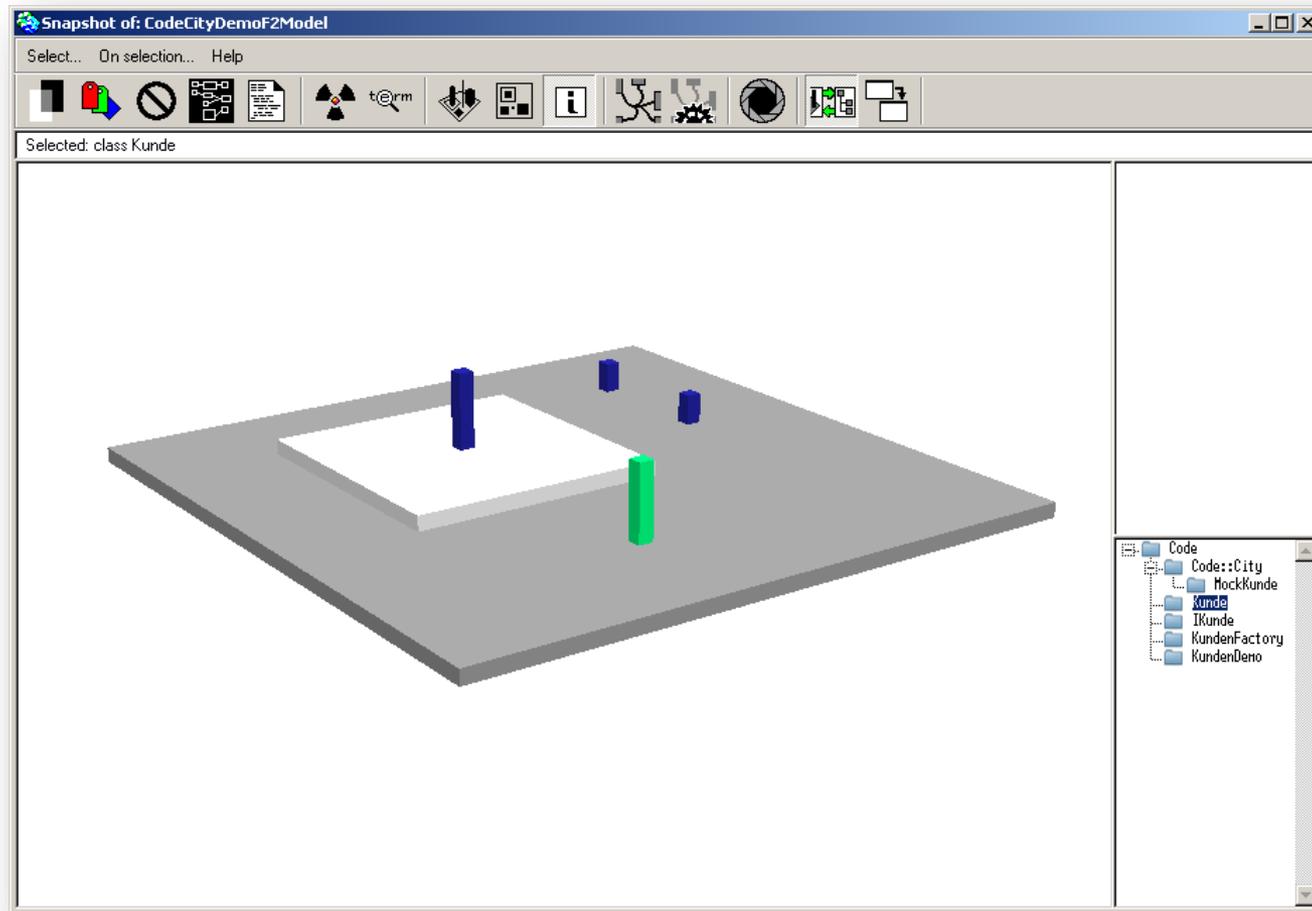


Vizz3D (2005)

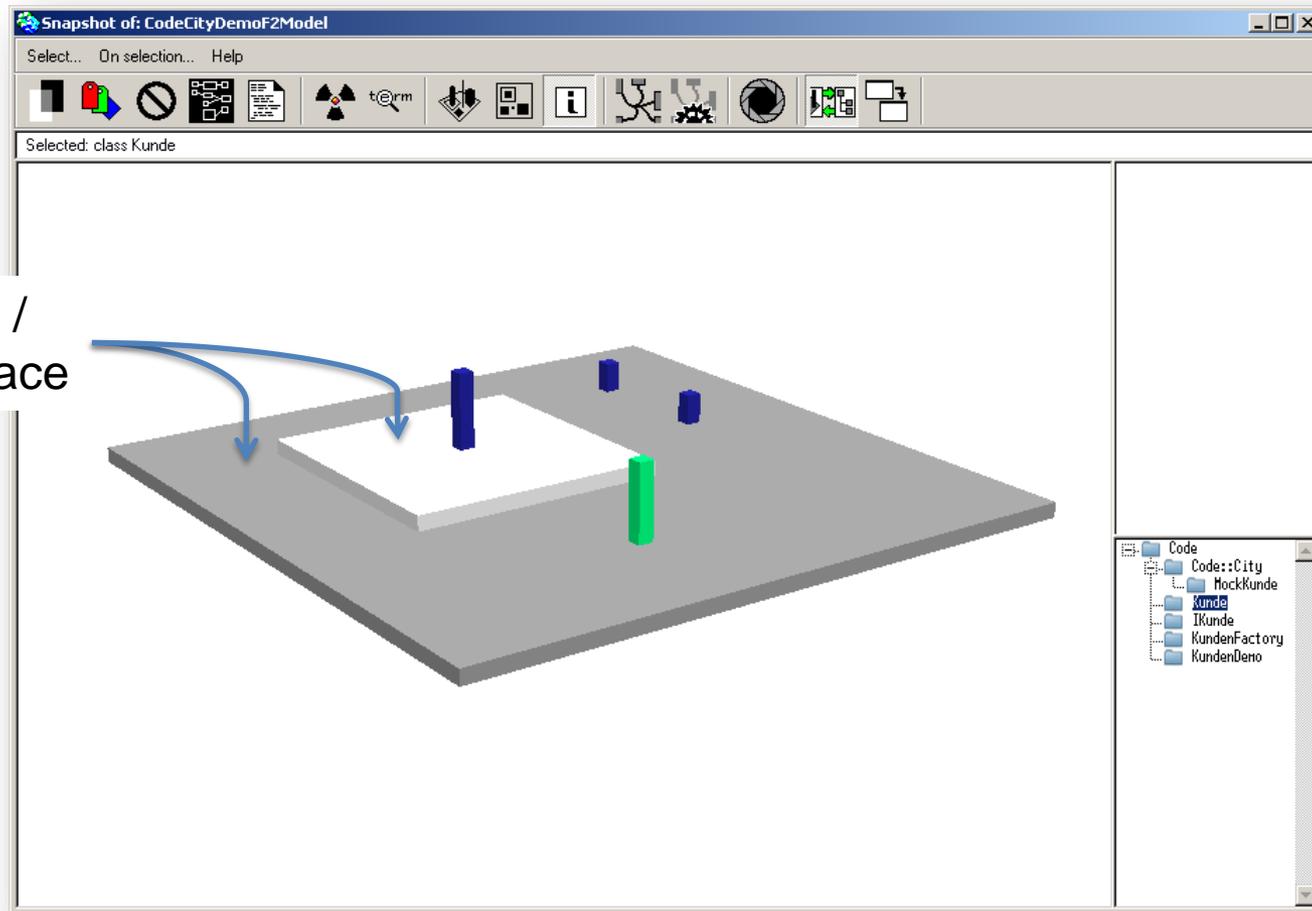
3D – CodeCity



3D – CodeCity

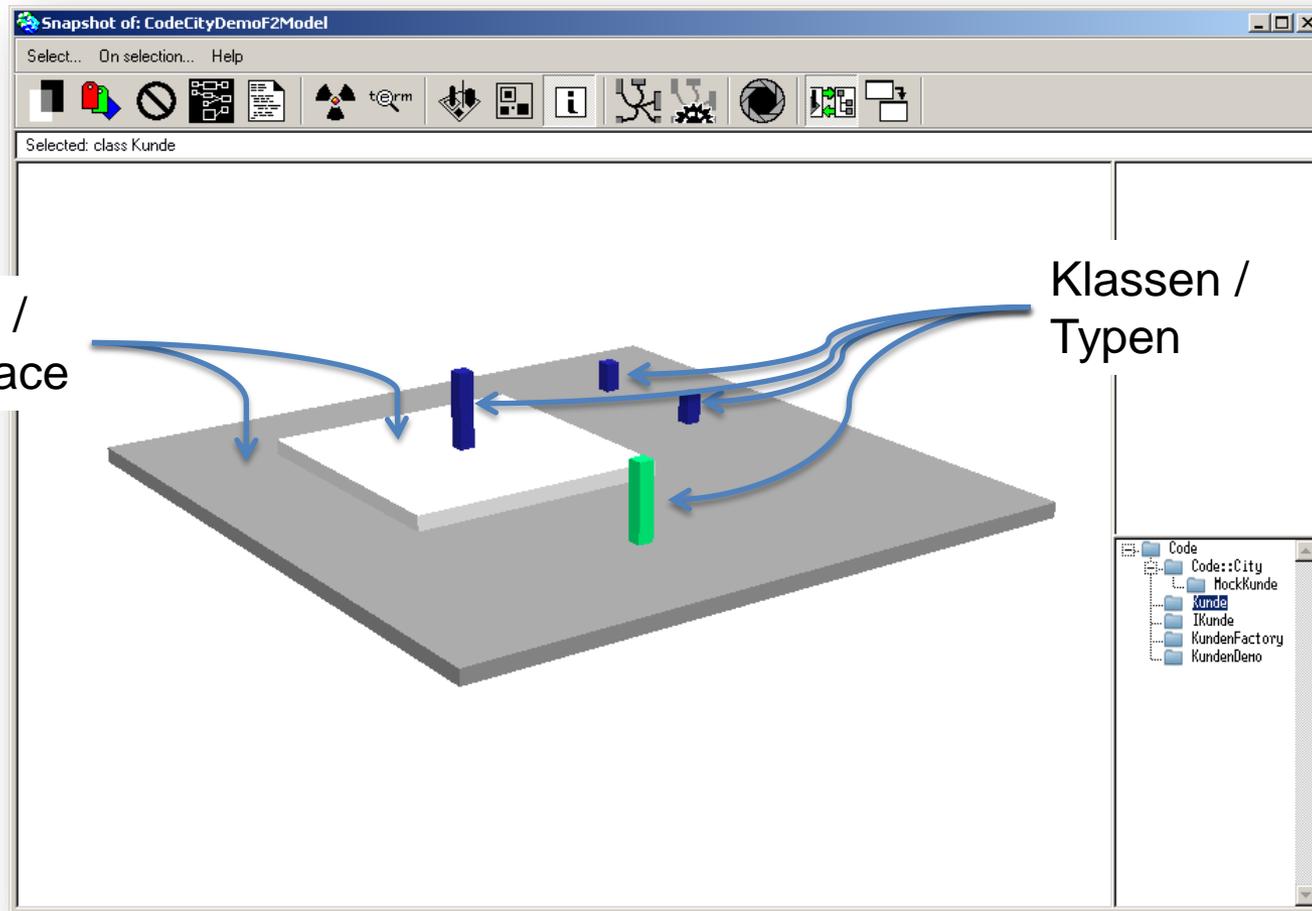


3D – CodeCity

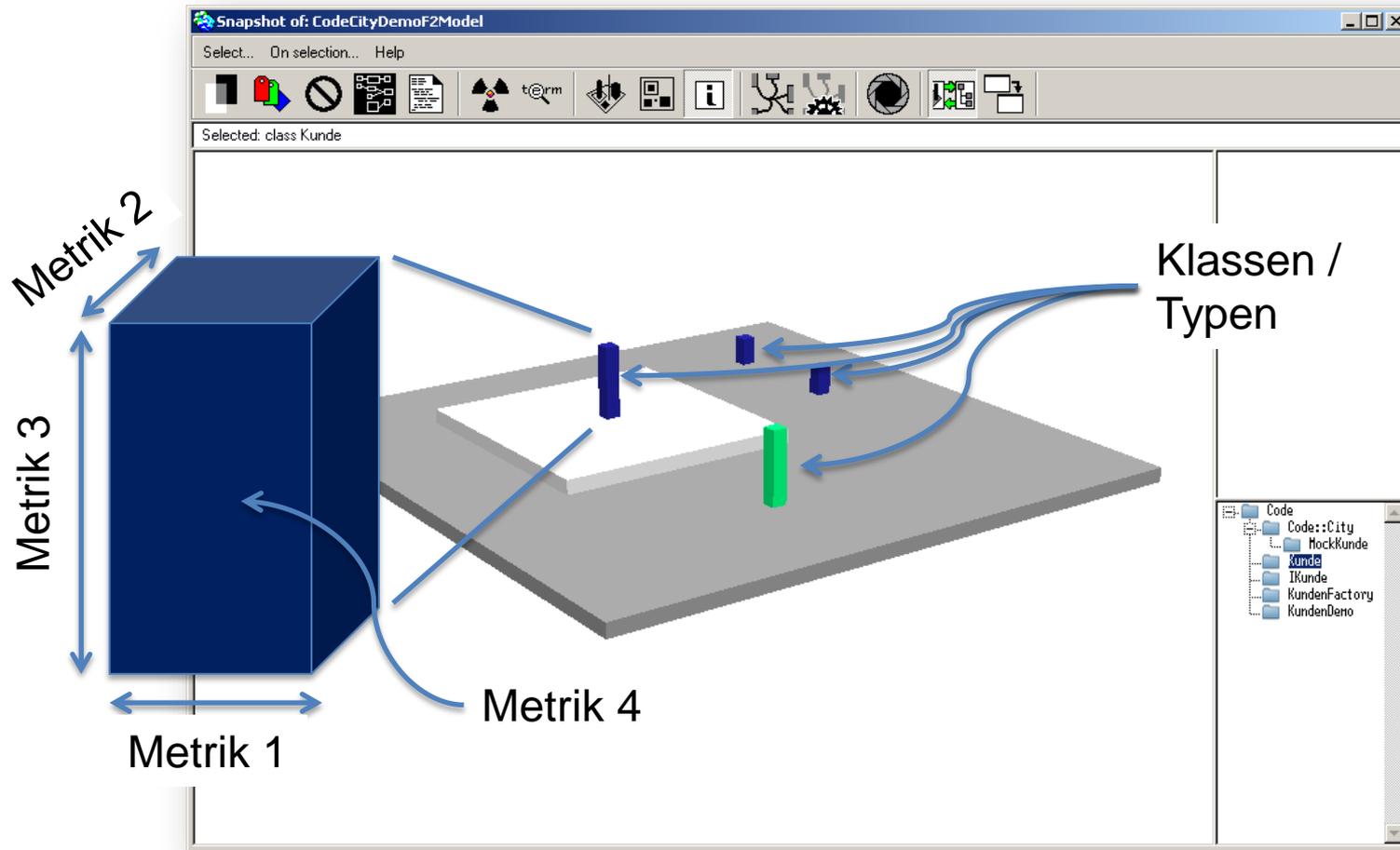


Package /
Namespace

3D – CodeCity

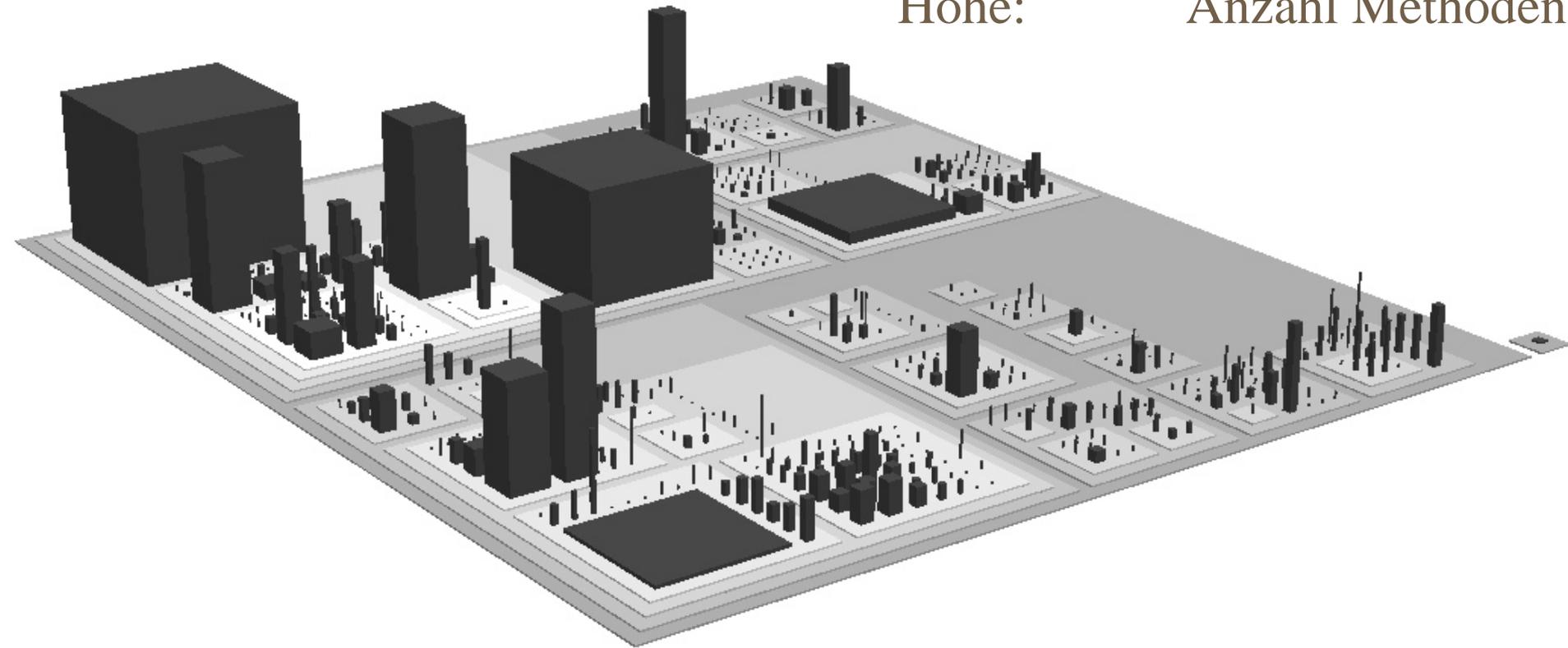


3D – CodeCity

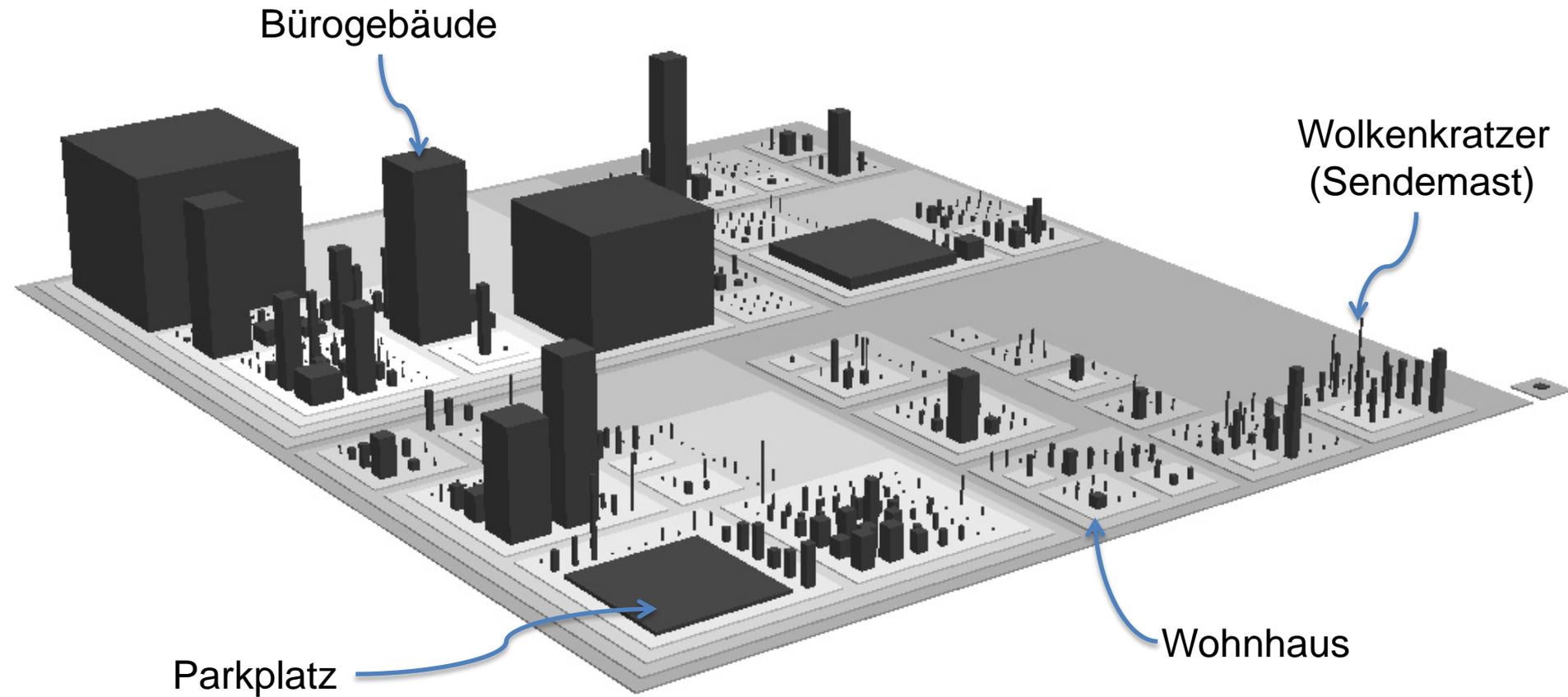


3D – CodeCity String.Net 1.3.1

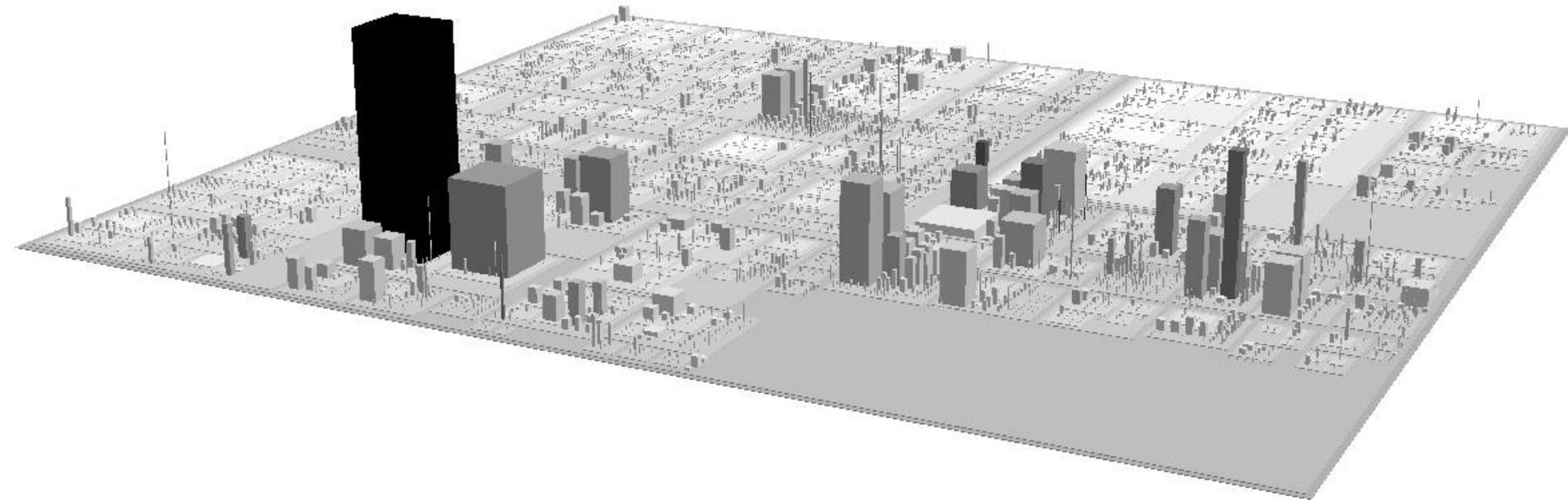
Grundfläche: Anzahl Attribute
Höhe: Anzahl Methoden



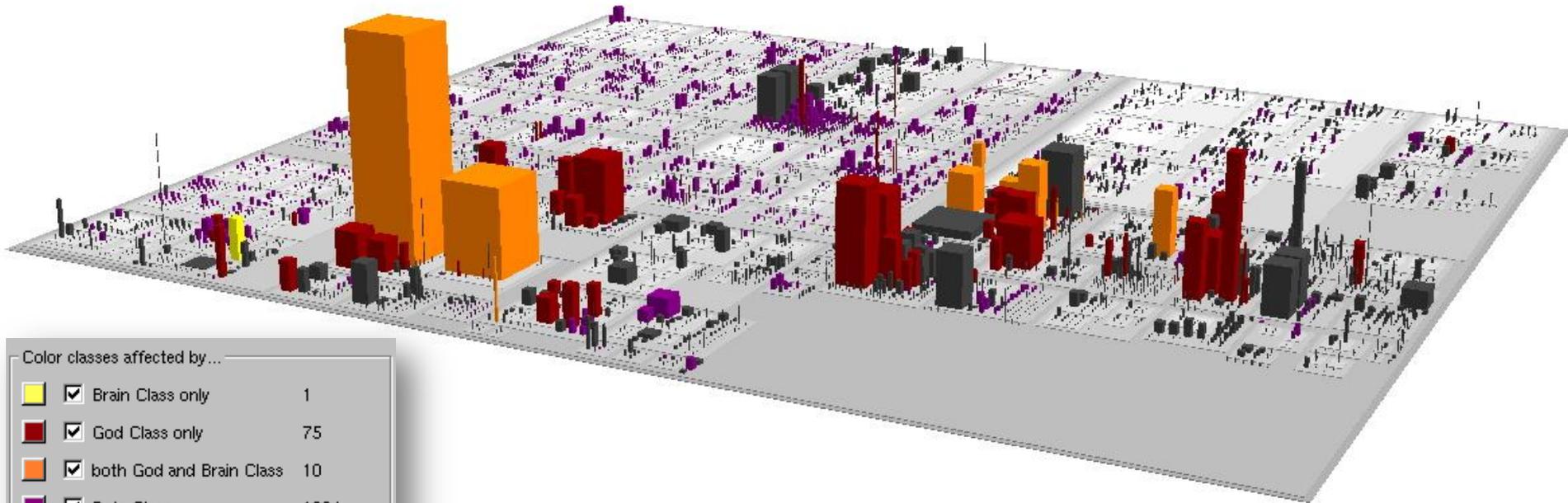
3D – CodeCity Gebäudetypen



3D – CodeCity Hibernate 3.5.6



3D – CodeCity *Disharmonies* Darstellung



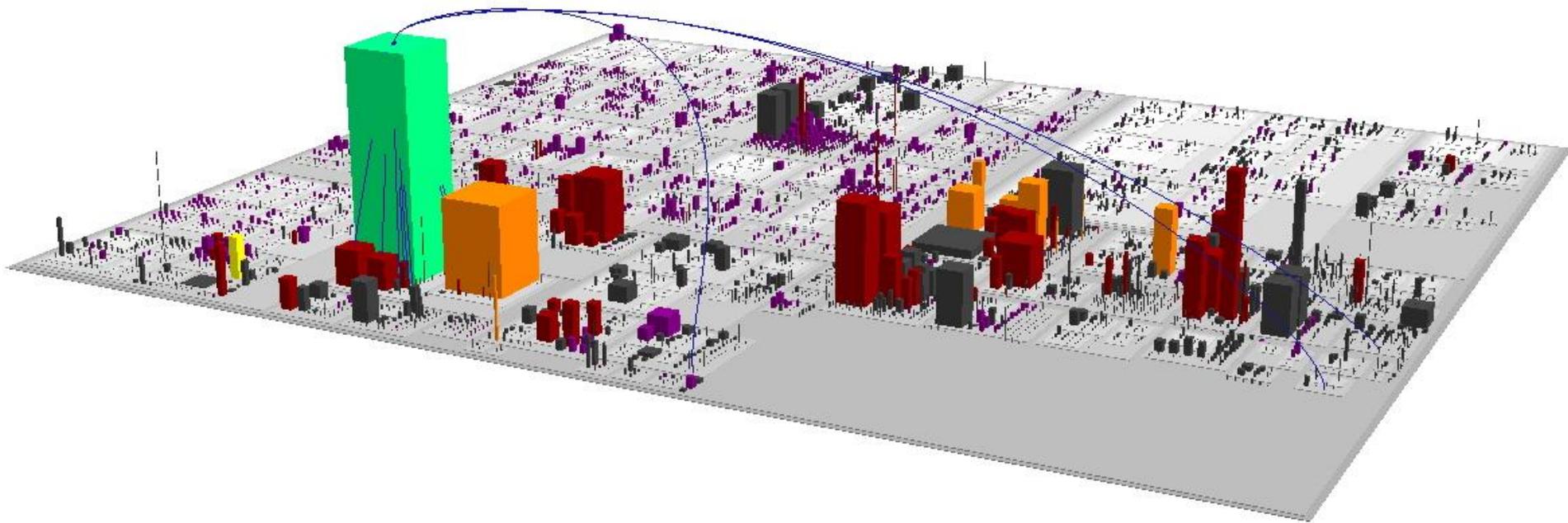
Color classes affected by...

	<input checked="" type="checkbox"/> Brain Class only	1
	<input checked="" type="checkbox"/> God Class only	75
	<input checked="" type="checkbox"/> both God and Brain Class	10
	<input checked="" type="checkbox"/> Data Class	1204
	<input checked="" type="checkbox"/> none of the above	3912

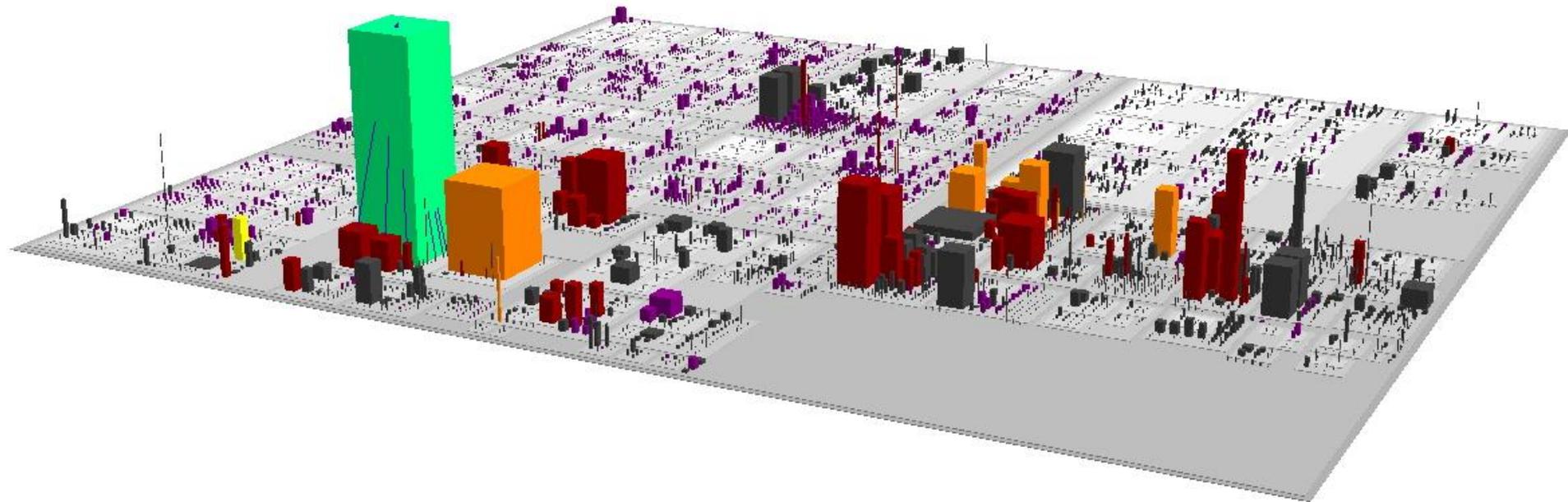
select all

unselect all

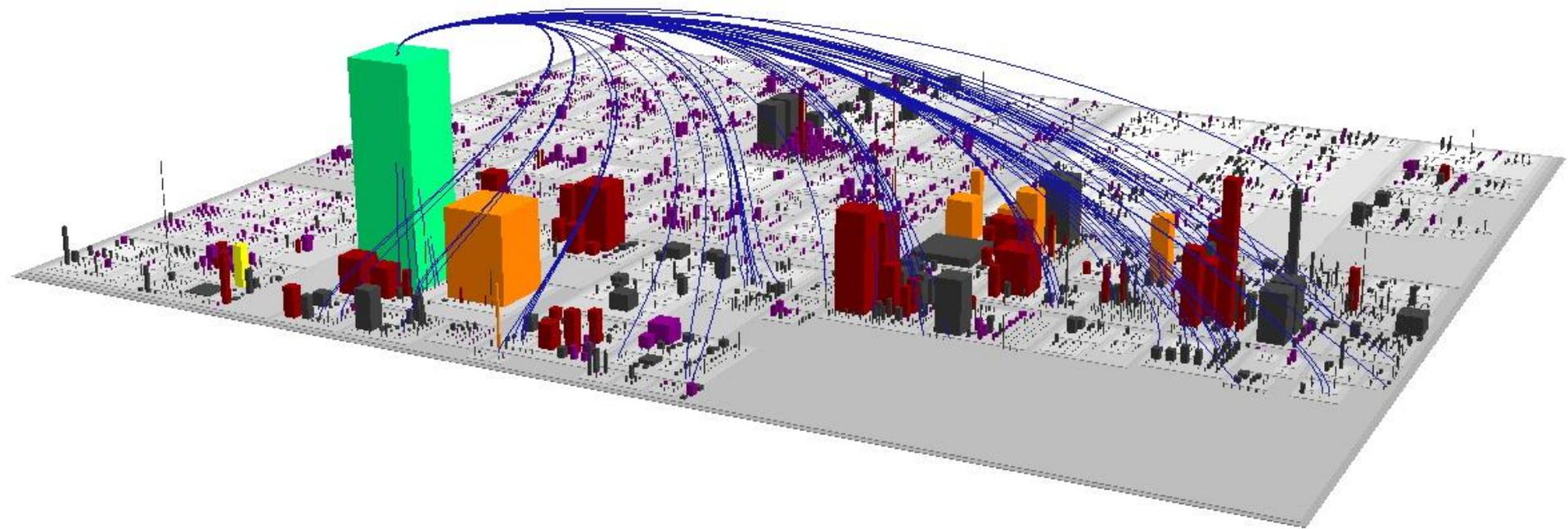
3D – CodeCity Vererbung



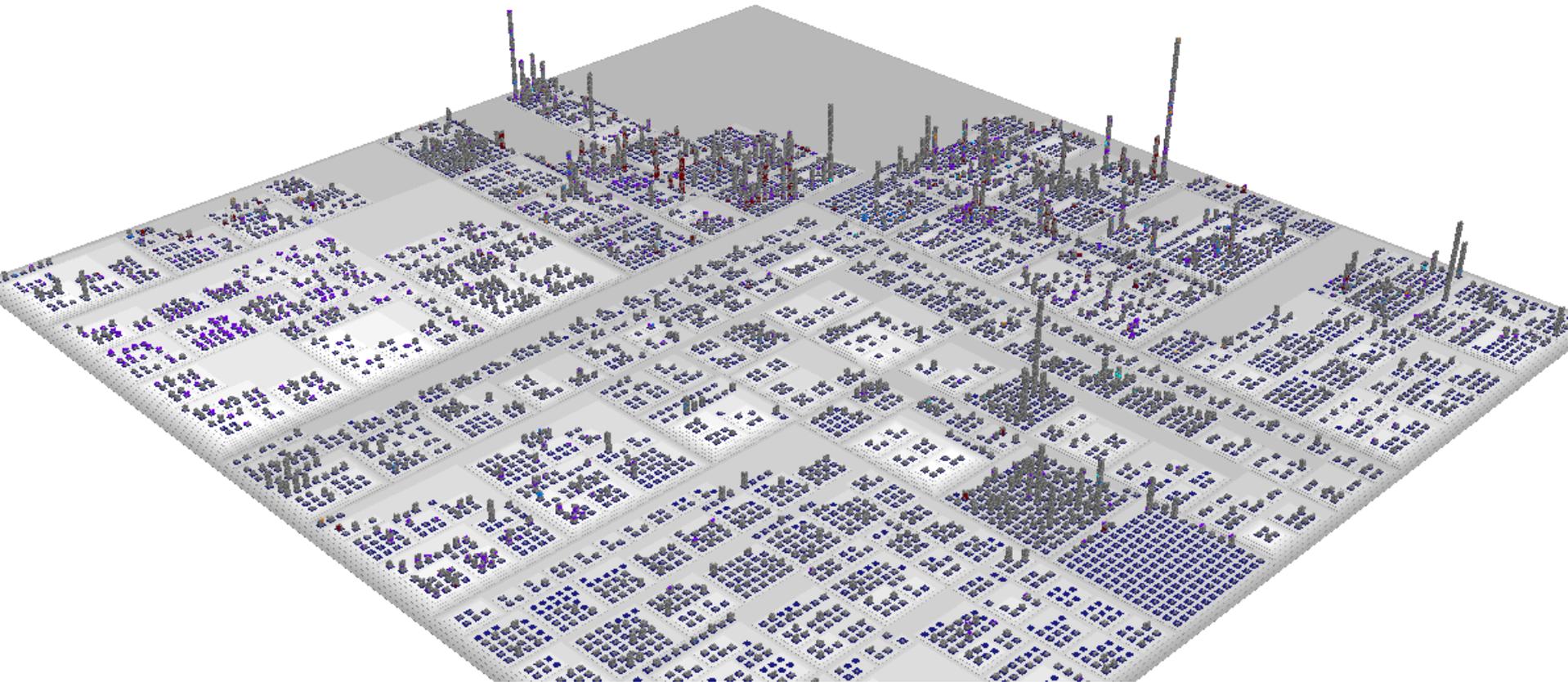
3D – CodeCity afferente Beziehungen



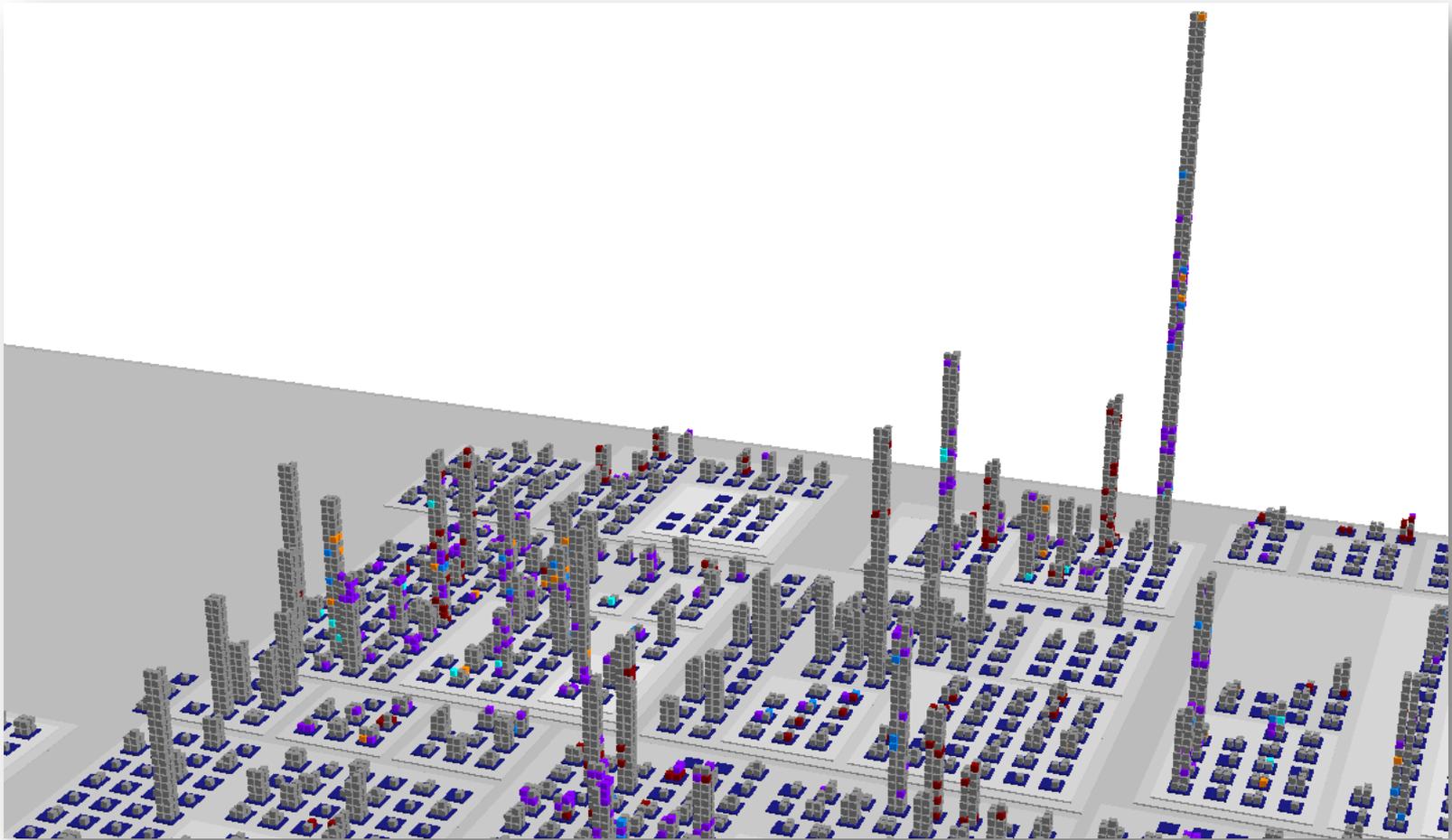
3D – CodeCity effereente Beziehungen



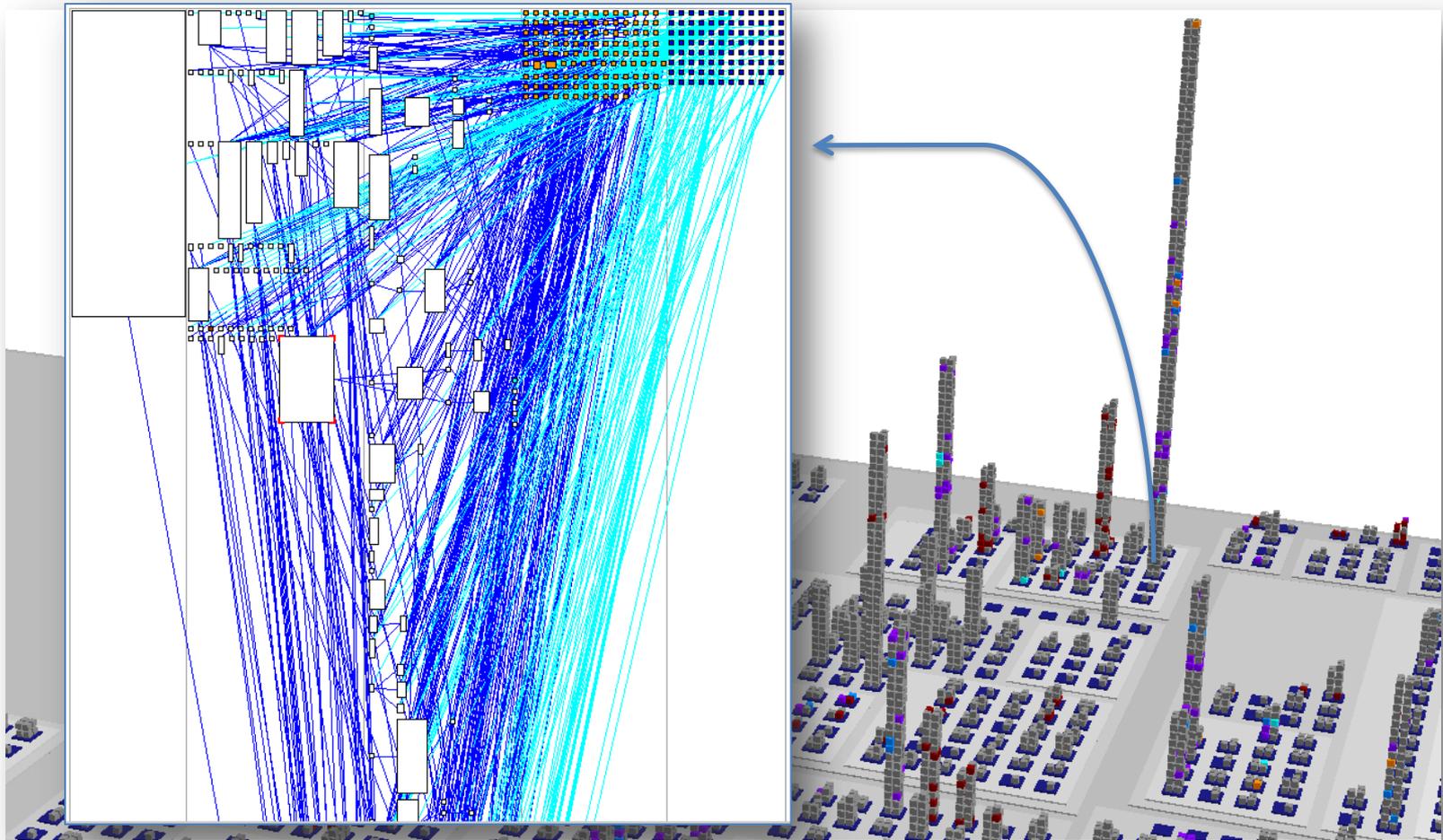
3D – CodeCity „Methodenansicht“



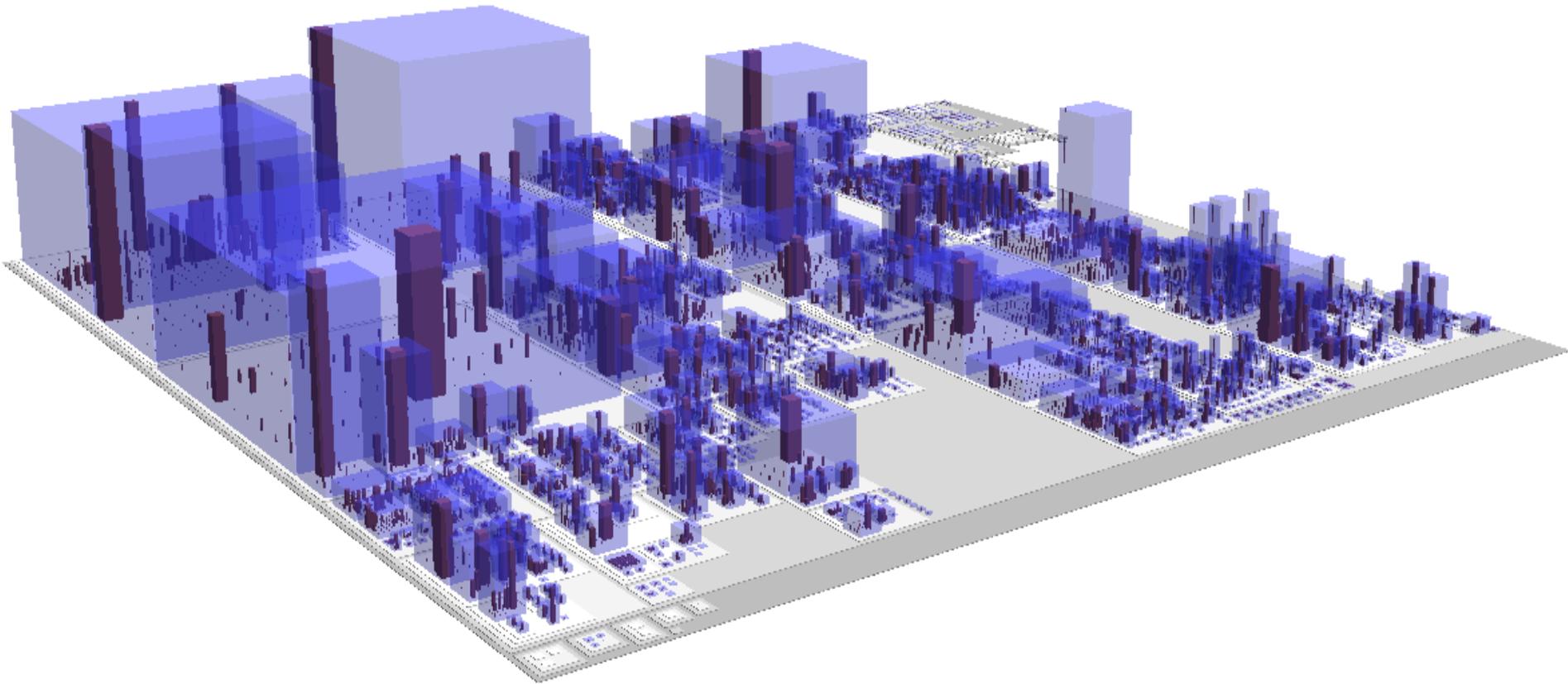
3D – CodeCity Methoden „*drilldown*“



3D – CodeCity Methoden „drilldown“



3D – CodeCity



Agenda

- Motivation

2D

- Overview first
- Zoom and filter
- Details-on-demand

3D

- Overview first
- Zoom and filter
- Details-on-demand

- **Zusammenfassung**

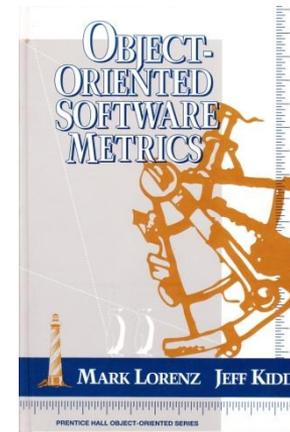
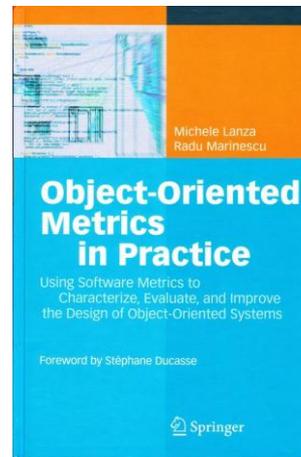
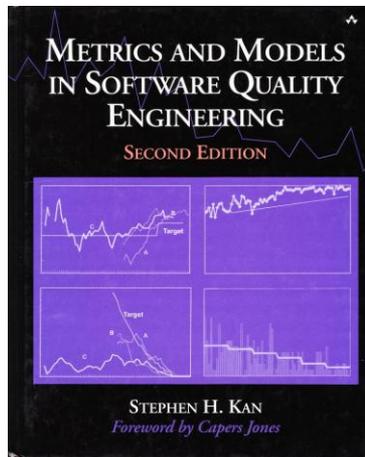
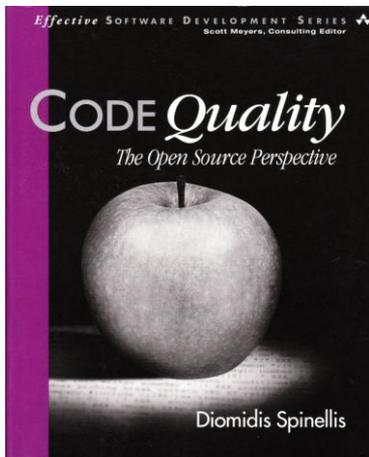
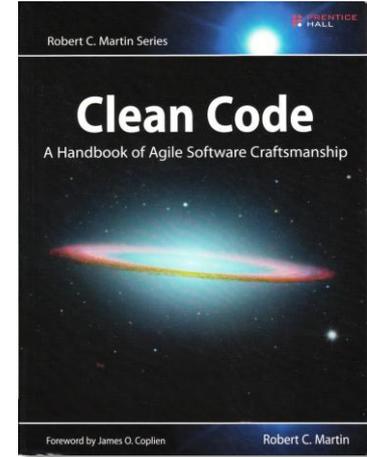
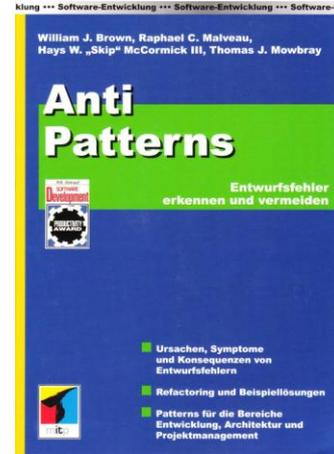
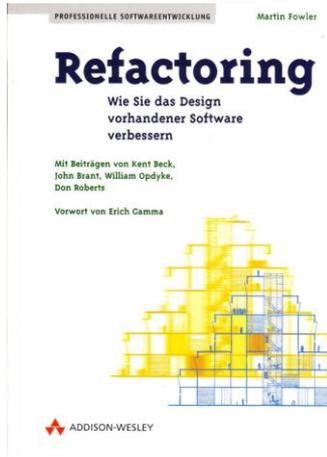
Zusammenfassung

- Grafische Darstellung von Metriken kann helfen
 - die Strukturierung zu verstehen (Pyramidal Overview)
 - die “hot-spots” des Systems zu identifizieren (Polymetric Views / treeView / Städte-Metapher)
- Die vorgestellten Werkzeuge haben alle ihre Vor- und Nachteile
- Weiterführende Analyse des Codes sollte dann „manuell“ erfolgen

Some tools

- ckjm <http://www.spinellis.gr/sw/ckjm/>
- PMD <http://pmd.sourceforge.net/>
- IPlasma <http://loose.upt.ro/iplasma/index.html>
- InFusion <http://www.intoitus.com>
- InCode <http://loose.upt.ro/incode/pmwiki.php/Main/Incode?from=Main.InCode>
- CodeCity <http://www.inf.unisi.ch/phd/wettel/codecity.html>
- MOOSE <http://moose.unibe.ch/>
- xRadar <http://xradar.sourceforge.net/>
- SonarJ <http://www.hello2morrow.com/products/sonarj>
- Sonar <http://www.sonarsource.org/>

Literatur



5.– 8. September 2011
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Vielen Dank!

Thomas Haug
MATHEMA Software GmbH