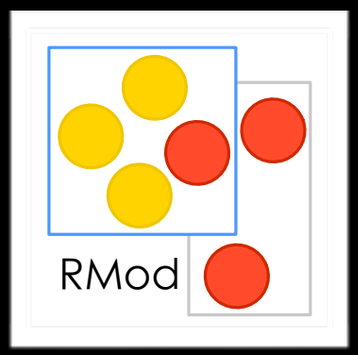
How worth is an end-user?

Platforms and Research:
The case of Pharo and Moose

http://stephane.ducasse.free.fr



Me in a Nutshell

- Inria Directeur de recherche
- Head of RMOD team (7 permanents, 20 people)
- 4 years scientific advisors of Inria Lille (300 people)
- 10 years Prof. in Software composition Group
- Wrote several open-source books
- Wrote many articles (h-index 47 - ~ 10000 citations)
- Leading the Pharo community http://www.pharo.org
- Co-funder of http://www.synectique.eu

synectique
Inventive Analysis

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

A kind of strange roadmap

- A thought about (bio) diversity
- Some questions about value
- Two Platforms
- Some Research @ RMOD

In biology diversity is a measure of ecosystem wealth

Endemic

adjective, Also, endemical

1.

natural to or characteristic of a specific people or place; native; indigenous

An endemic species is one whose habitat is restricted to a particular area.

Socrota Island

As of 1990, Socrota counted
700 endemic species

As of 1999, England counted 47 floral endemic species

Which one would you keep and preserve?

Even without a french perspective … we would preserve Socrota :)

Why we do not value diversity in CS research?


Why should we all follow
the trend to be relevant?

In 2002 I taught JavaScript

but I was not cool

In 2014 I do not teach
JavaScript
but I’m still not cool ;)

Back in 1998, Java was such a dogma that we (W.De Meuter, T. D'Hondt and Ole Madsen) created of the "OO Language engineering in Post Java-Area" Int. Workshop

Lambda the Ultimate

The Programming Languages Weblog

OO Language Engineering for the Post-Java Era

> *... Java also acted as a brake especially to academic language design research... The goal of this second edition of the workshop was to address object-oriented languages that diverge from Javaâ€™s doctrine but support a much more dynamic way of constructing software. In the near future, this dynamicity will be required in order to construct software that is highly context-dependent due to the mobility of both the software itself and its users...*

ECOOP 2004 Workshop - Back to Dynamicity

ECOOP 2003 Workshop

By Isaac Gouy at 2005-09-02 19:32 | LtU Forum | previous forum topic | next forum topic | other blogs | 8402 reads

How did we succeed to escape from Cobol?

How can we influence that the next language will exhibit different properties if we do not explore different approaches?

“your approach is interesting but it does not apply to Java”

“your approach is interesting but it does not apply to Java”

luckily

“your approach is interesting
but it does not apply to Java”

luckily
but often wrong statement,of
course :)


WHAT ARE YOU DOING?! YOU'RE GOING TO BE LATE FOR SCHOOL! HURRY UP AND PUT YOUR CLOTHES ON RIGHT!


IT'S SAD HOW SOME PEOPLE CAN'T HANDLE A LITTLE VARIETY.

When we evaluate other researchers…

We count…

Papers

Papers

Papers

ah yes also

PC participation and keynotes :)

also **boring**
no impact
editorial board and
admin tasks

This is cannot be the only measures

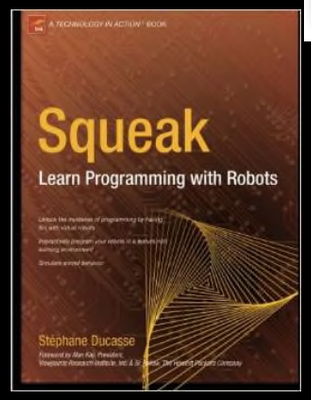
How many papers is worth one user?

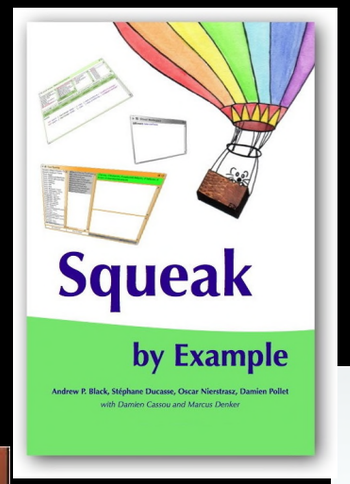
(Not one of our students!
Just a real end user)

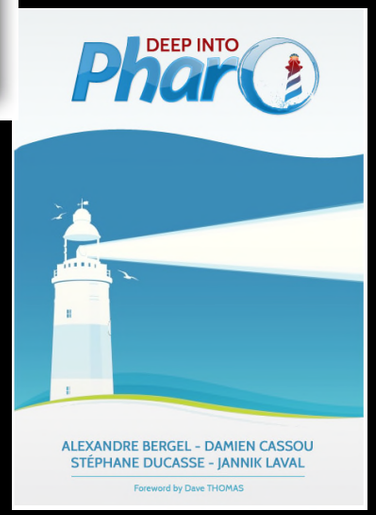
How many papers is worth one real user?

How many papers is worth a book for normal people?

Normal people do not refer to your work (they use it) - no citation carrot





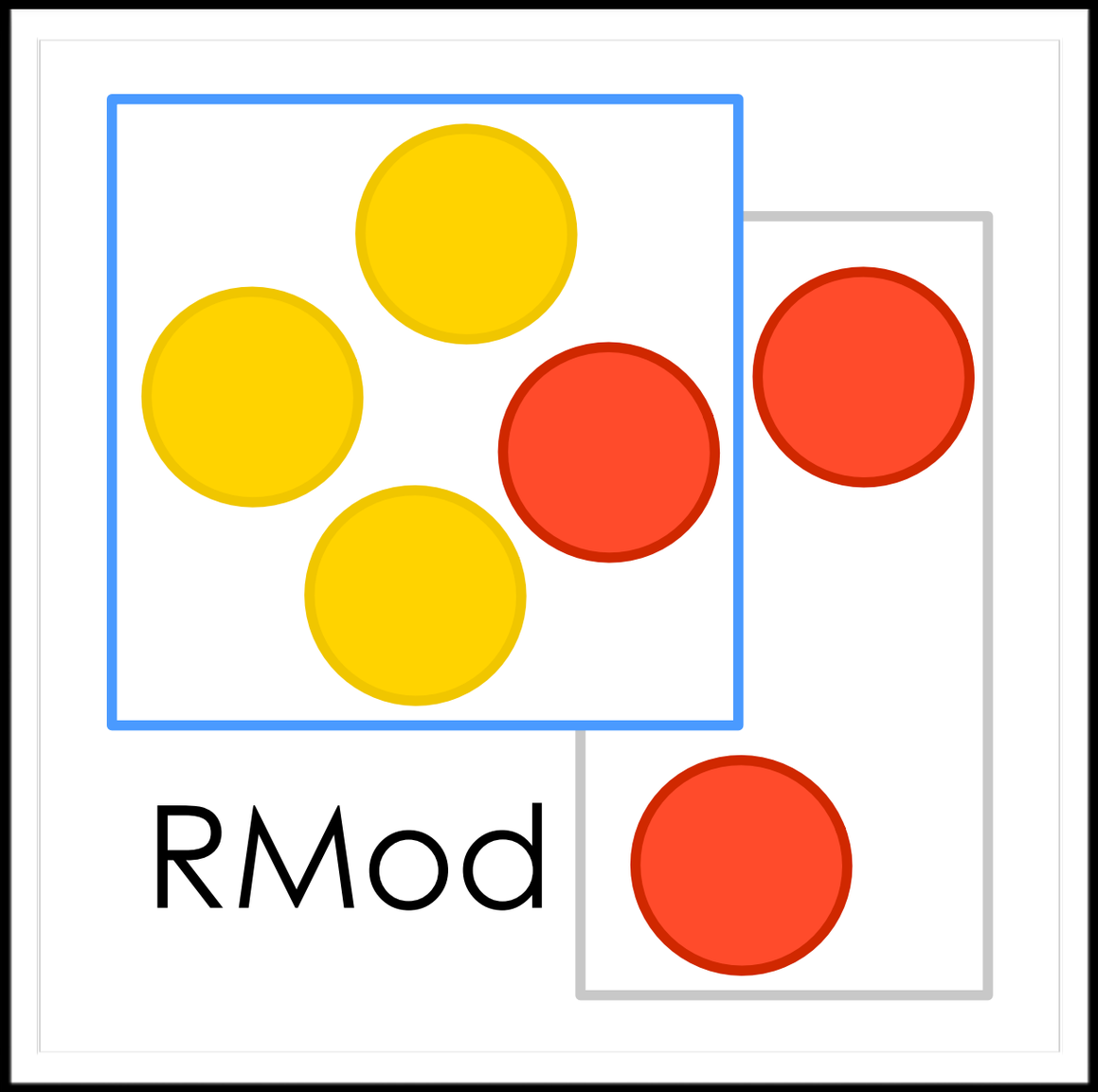


(Designing this keynote
I had a thought about
DrScheme/Racket guys!
I value your work, guys!)

How many papers is worth a startup?

I just have the questions
you have the answer
:)


Stop ranting!
Now the fun


RMod

Created in 2009

Permanent staff

- S. Ducasse, DR2 INRIA
- D. Pollet, MCF Lille
- N. Anquetil, MCF Lille
- M. Denker, CR1 INRIA
- D. Cassou, MCF Lille
- A. Etien, MCF Lille

Non-permanent staff

- 1-2 Engineers
- 5-7 PhDs
- 1-2 PostDocs
- 3-5 interns

We are hiring!

Objectives

- **Objective 1: Reengineering**

  How to maintain/evolve large software systems?

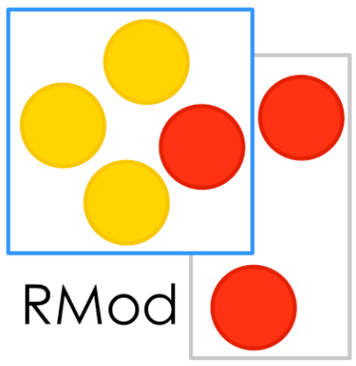
- **Objective 2: Supporting evolution and isolation**

  Revisiting fundamental aspects of OO languages

- **Objective 3: Ecosystem around Pharo**

  Platform used to create wealth and innovation

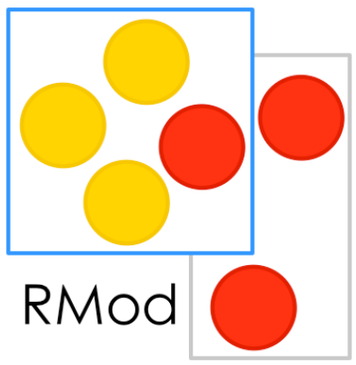
Synergy



Reengineering

Languages & IDE

Synergy



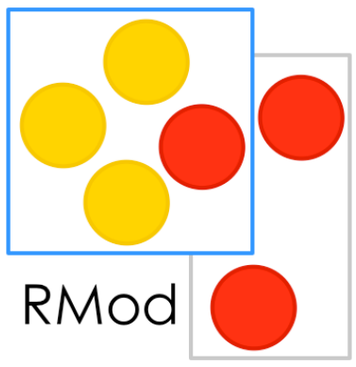
Reengineering

- 230 KLoC
- 2400 classes

Languages & IDE

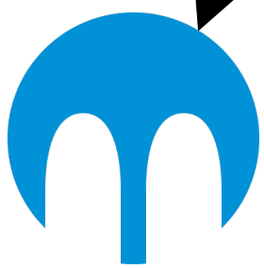
- 500 KLoC
- 5000 classes

Synergy



Reengineering

Languages & IDE

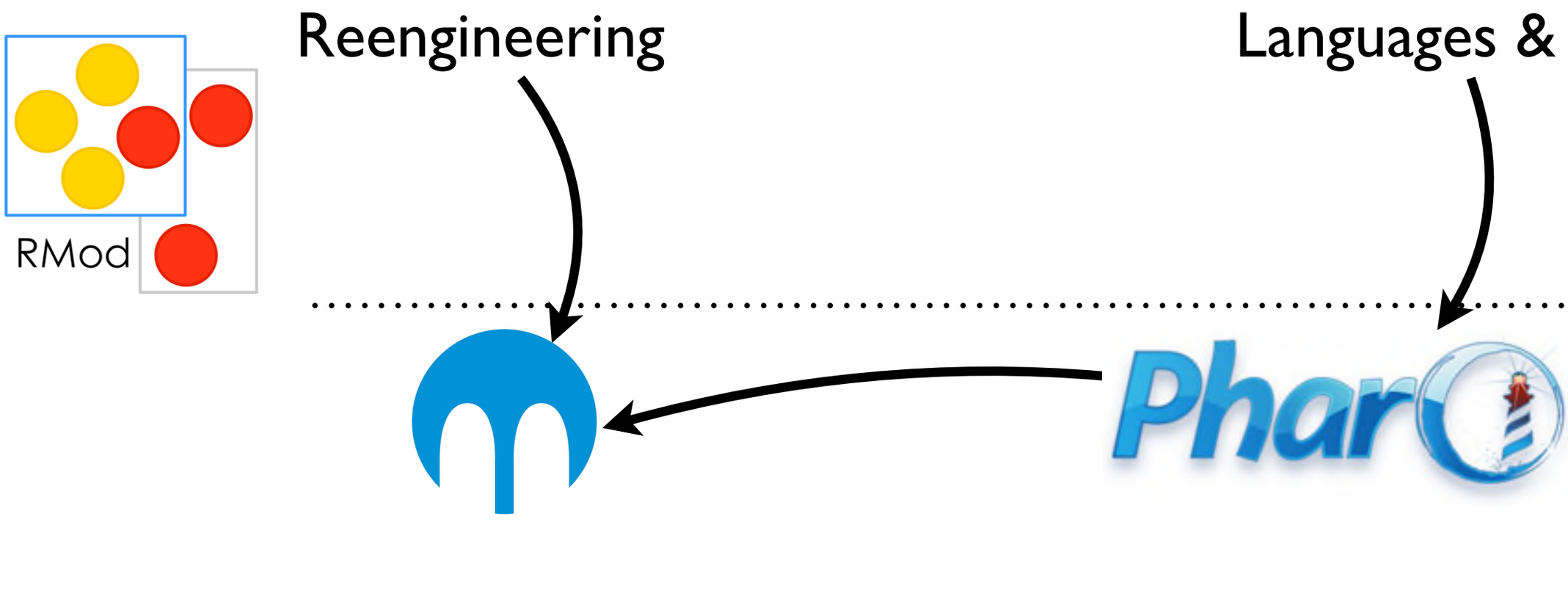


Moose 5.1

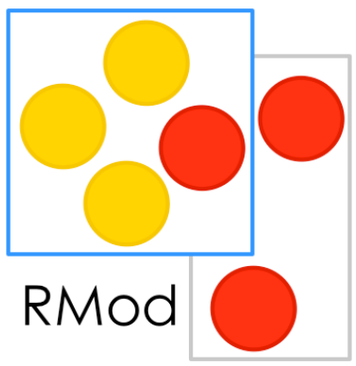
Pharo 4.0 released last April

- 1700 issues closed
- replacement of 3 core tools
- 1st-class instance variables (slots)
- new GUI framework (Glamour)
- …

Synergy

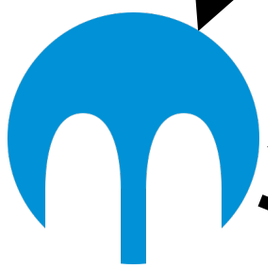


Synergy



Reengineering

Languages & IDE

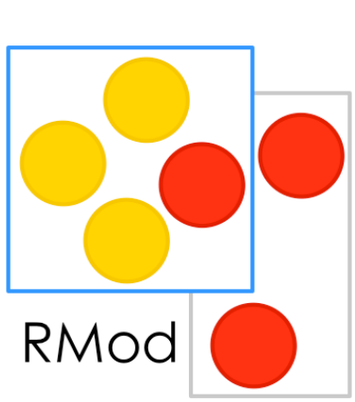


27
universities

15
research
groups

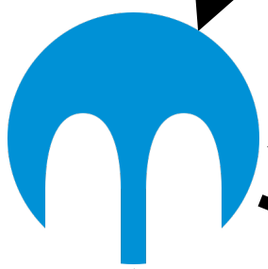
50 companies

Synergy



Reengineering

Languages & IDE

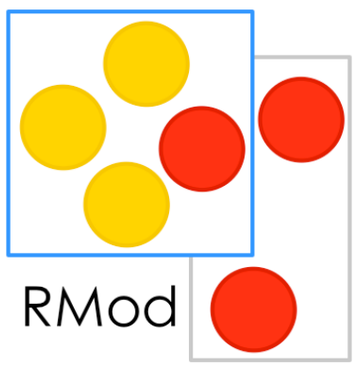


27 universities

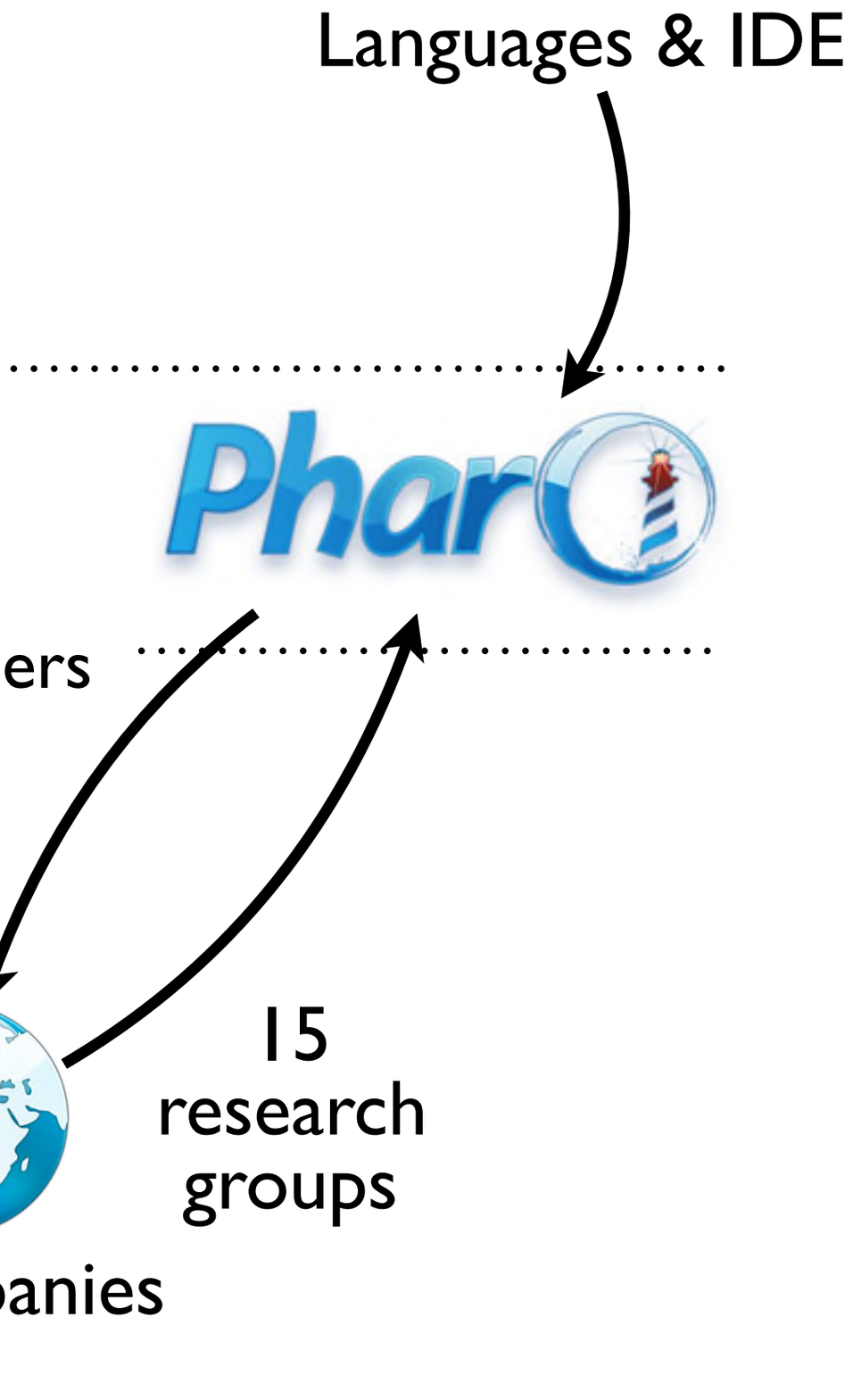
15 research groups

50 companies

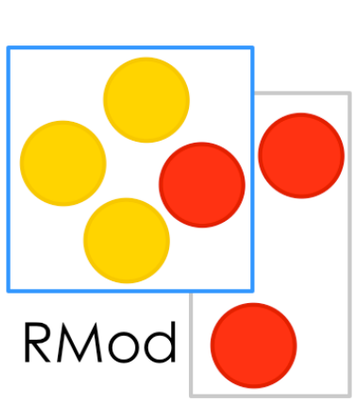
Synergy



- 77 contributors to Pharo 4.0
- 215 license agreements
- 70 association members
- 13 industrial consortium members

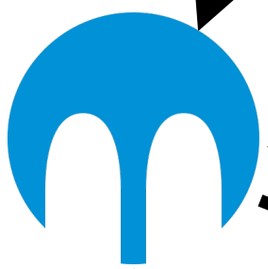


Synergy



Reengineering

Languages & IDE

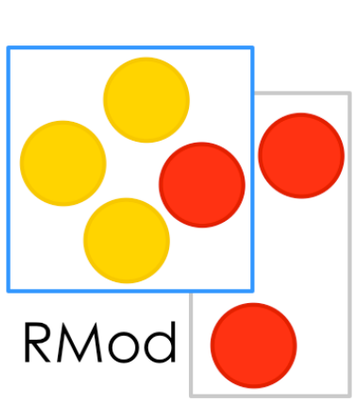


23
universities

13
research
groups

50 companies

Synergy



Reengineering

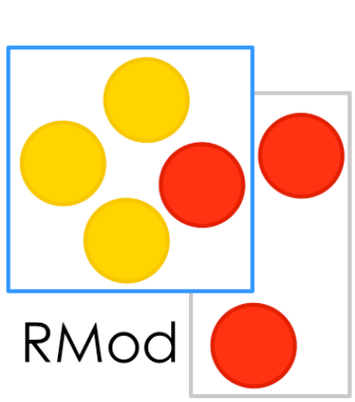
Languages & IDE

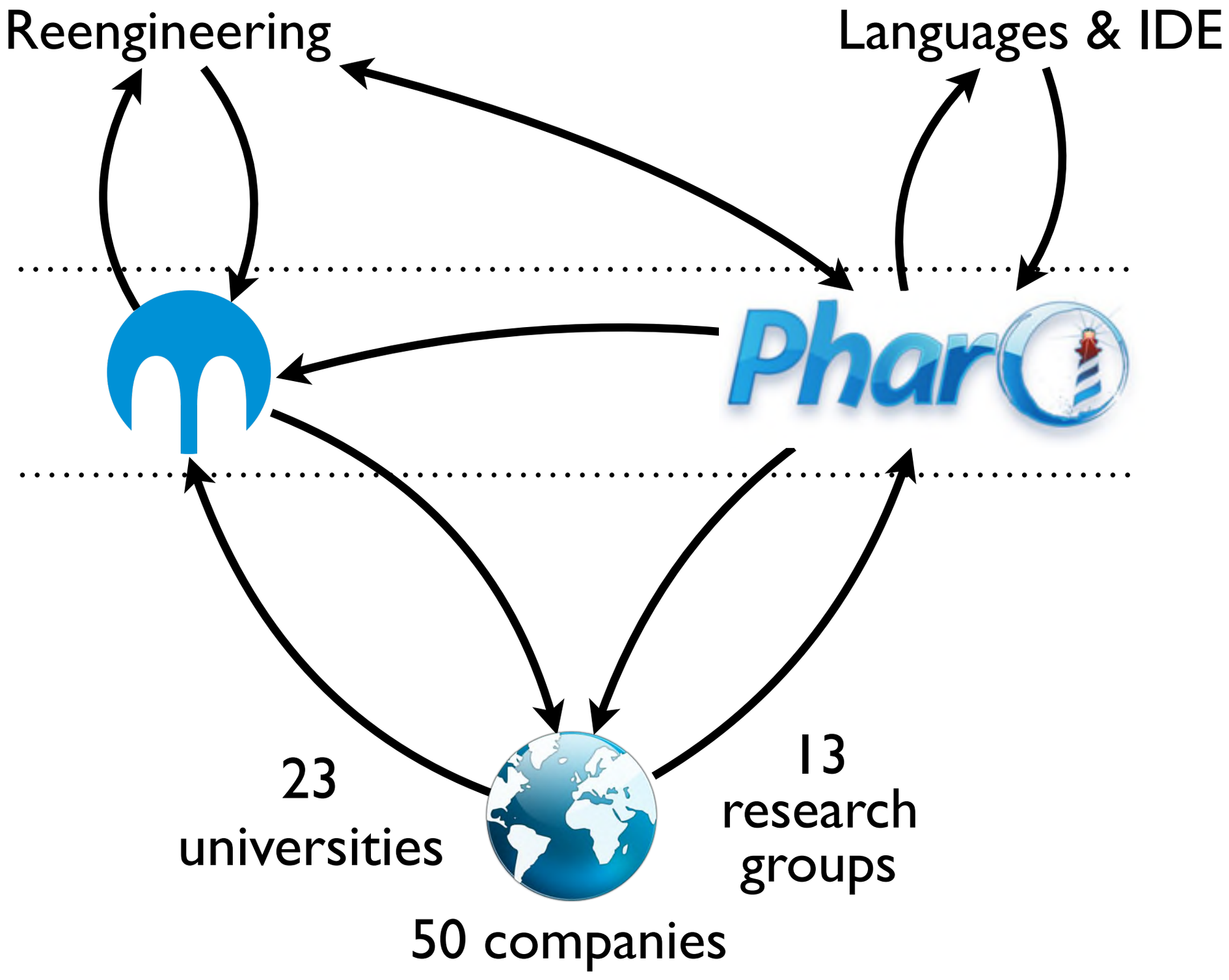
23
universities

13
research
groups

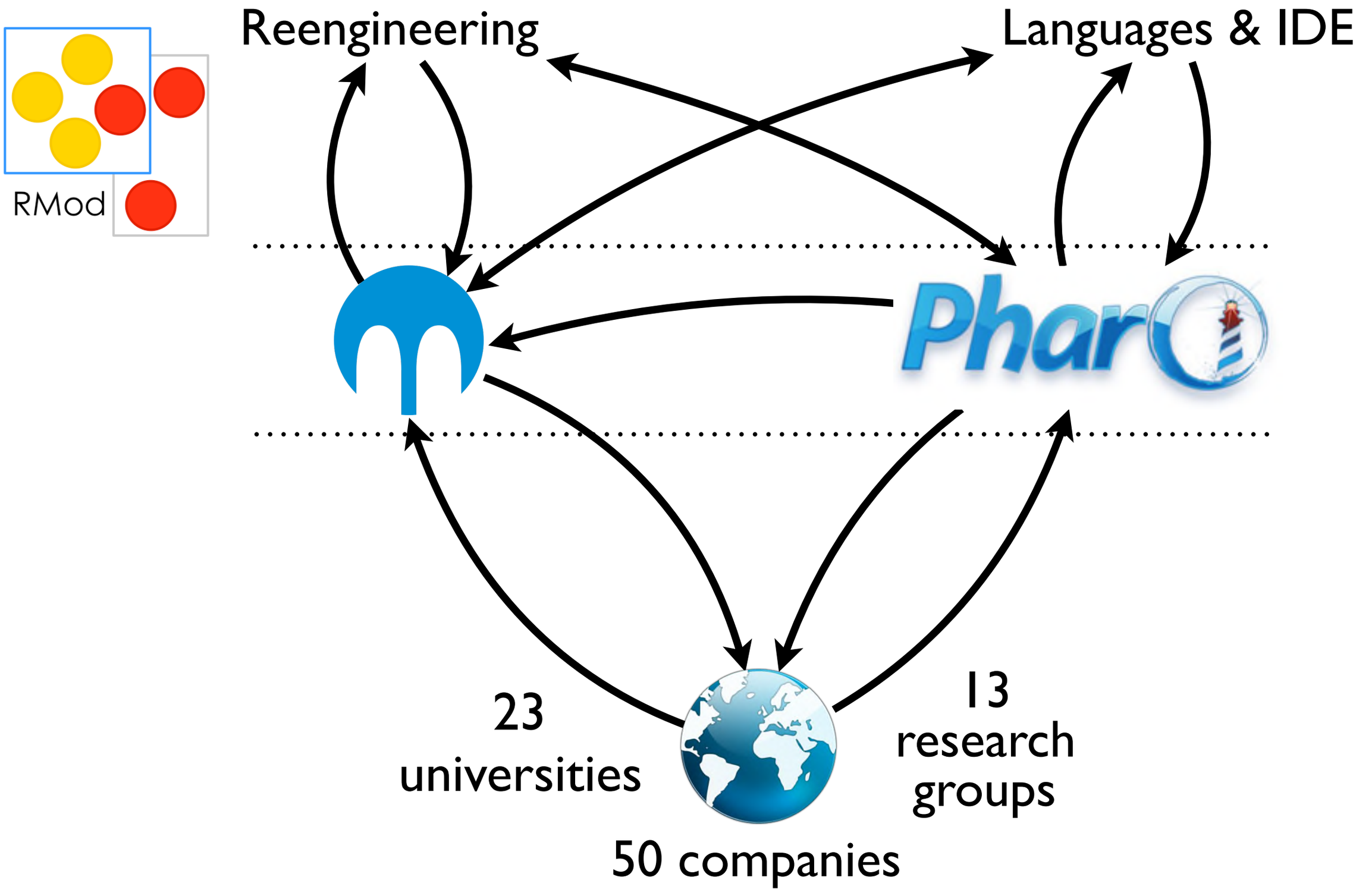
50 companies

Synergy


RMod


Reengineering
Languages & IDE
Pharo
23
universities
13
research
groups
50 companies

Synergy


RMod
Reengineering
Languages & IDE
Pharo
23
universities
13
research
groups
50 companies

Pharo?

Pharo!

- http://www.pharo.org
- System: Pure object language + full IDE
- Inspired by Smalltalk - fork in 2008 of Squeak
- Powerful, elegant and fun to program
- Great community
- Living system under your fingers

Elegant!

- Full syntax on a postcard
- Simple but powerful object model

Complete Syntax on a Postcard

```
exampleWithNumber: x
    "A method that illustrates every part of Smalltalk method syntax"
    <menu>
    | y |
    true & false not & (nil isNil) ifFalse: [self halt].
    y := self size + super size.
    #($a #a 'a' 1 1.0)
        do: [ :each | Transcript
                show: (each class name);
                show: (each printString);
                show: ' '].
    ^ x < y
```

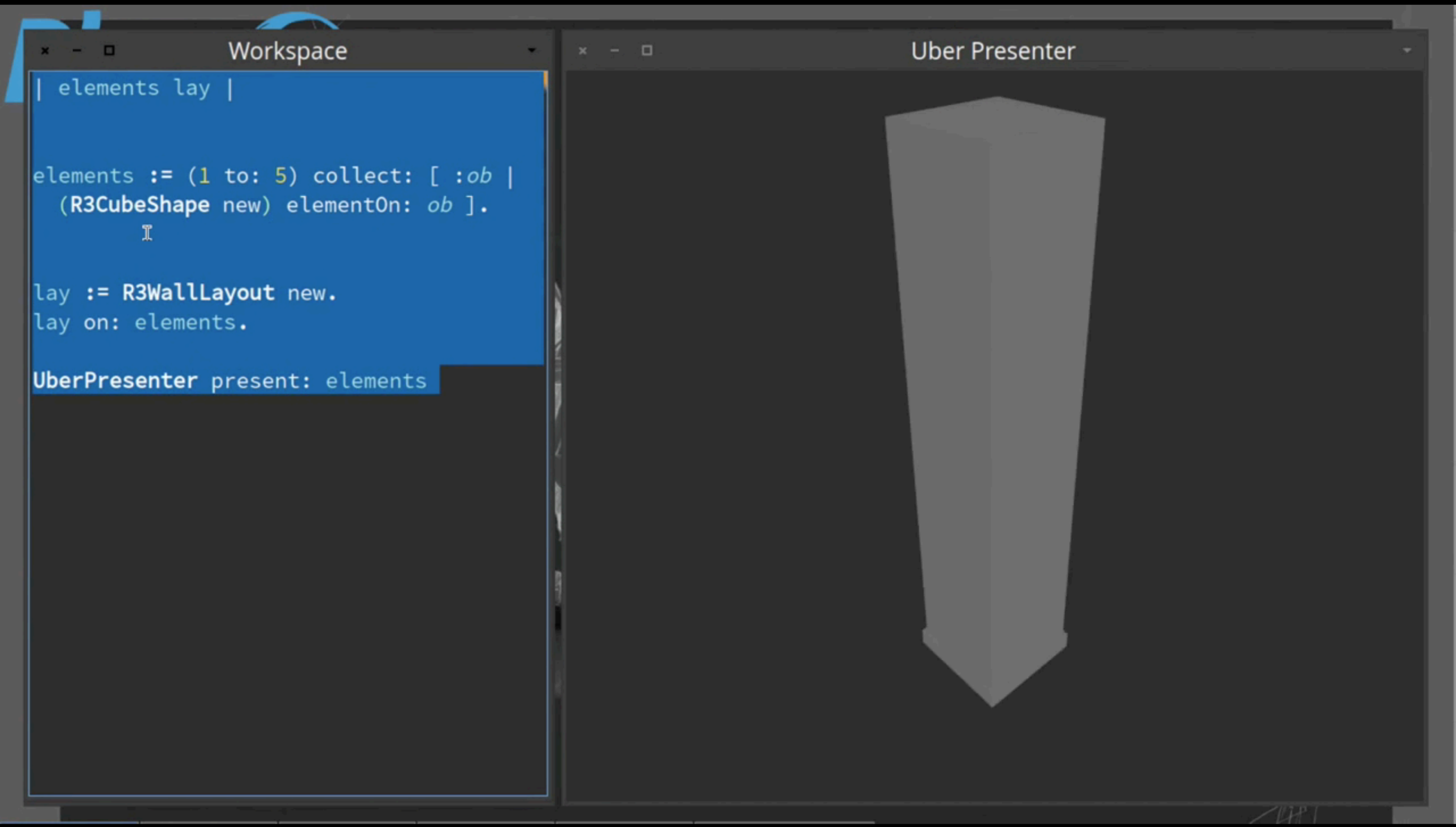
Object Model

- Dynamically typed
- **Everything** is an instance of a class
- All methods are public and virtual
- Attributes are protected
- Single Inheritance

Immersive?

Immersive!

- Deep contact with objects
- Highly interactive programming sessions
- Reflective, inspectable



http://youtu.be/CuimMwuZiGA

Workspace

```
| elements lay |


elements := (1 to: 5) collect: [ :ob |
  (R3CubeShape new) elementOn: ob ].


lay := R3WallLayout new.
lay on: elements.

UberPresenter present: elements
```

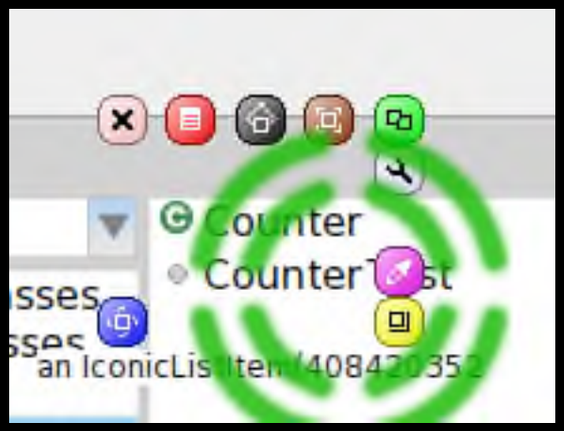
Uber Presenter

Any object can be lively updated

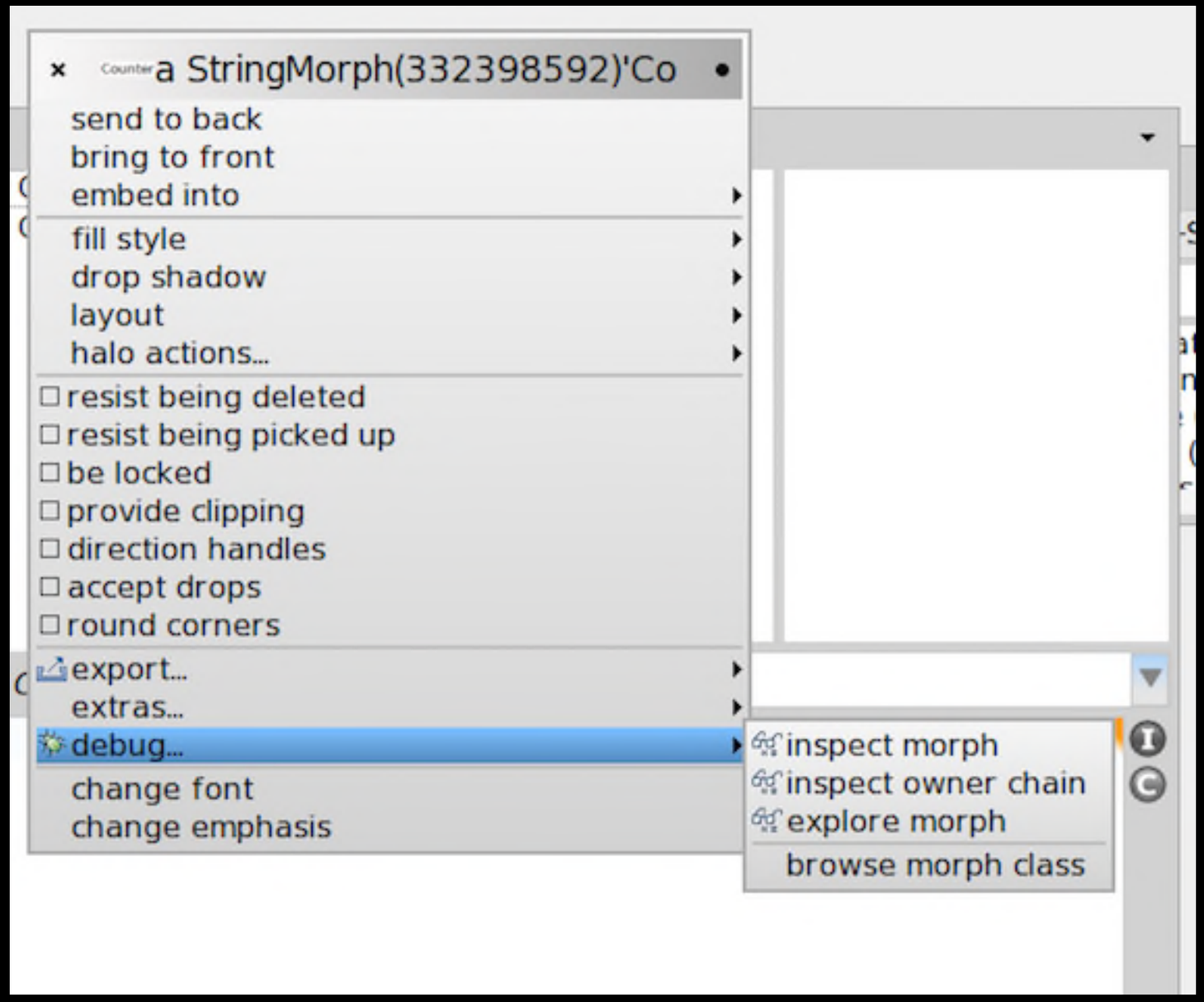
- 3D
- Network connection
- Dynamic web applications
- GPU
- Code

Learning from the system...

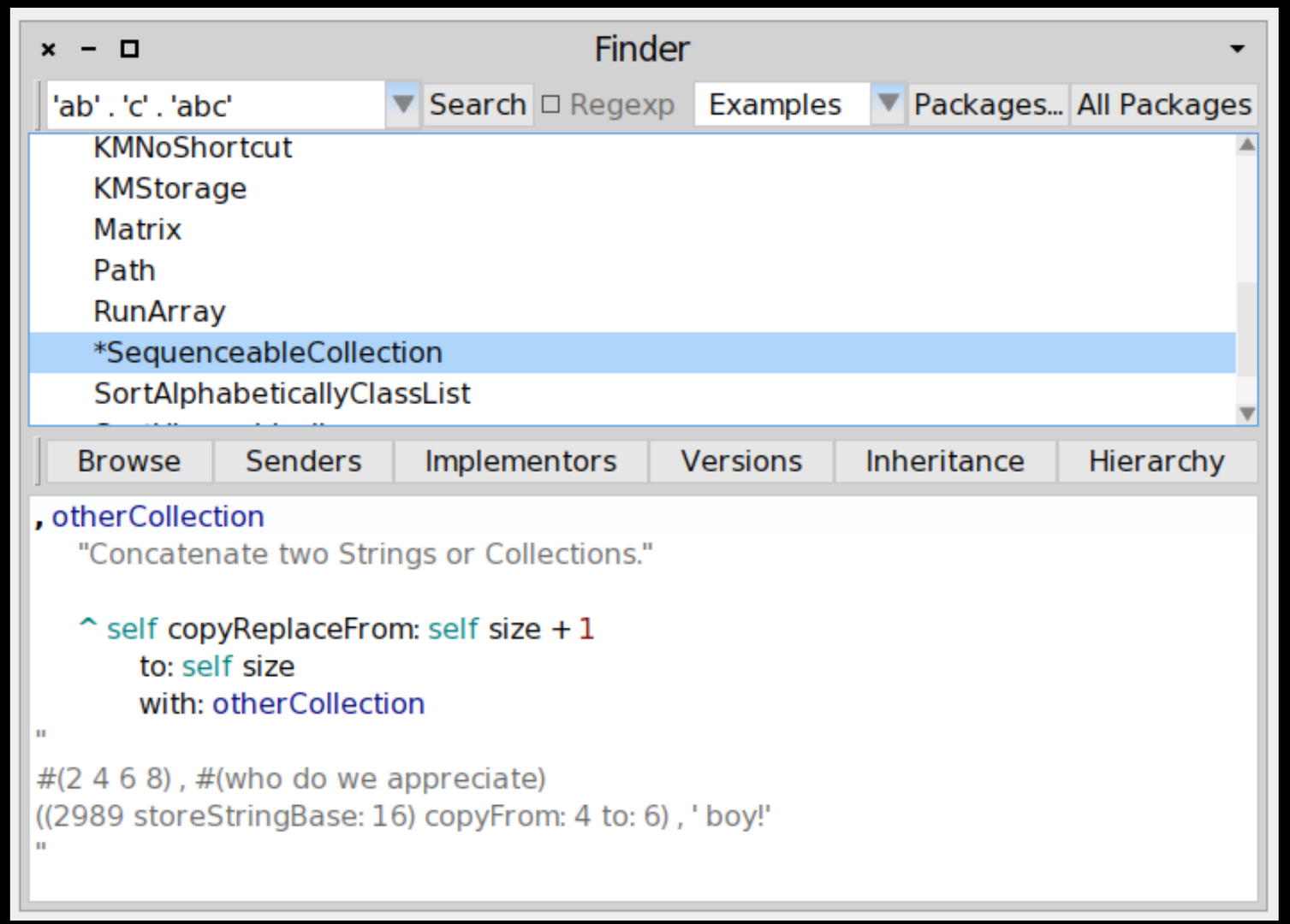
Click on it :)
Cmd+shift+option



- Cmd-Shift+option



give examples and
get the methods that works!



A system to learn
advanced
oo design

How to implement not?

- false not -> true
- true not -> false

Let the receiver decide!

Not implementation

```
False>>not
    ^true

True>>not
    ^false
```

Implementors of not [9]

Boolean (logical operations)	not	[Kernel]
False (logical operations)	not	[Kernel]
True (logical operations)	not	[Kernel]

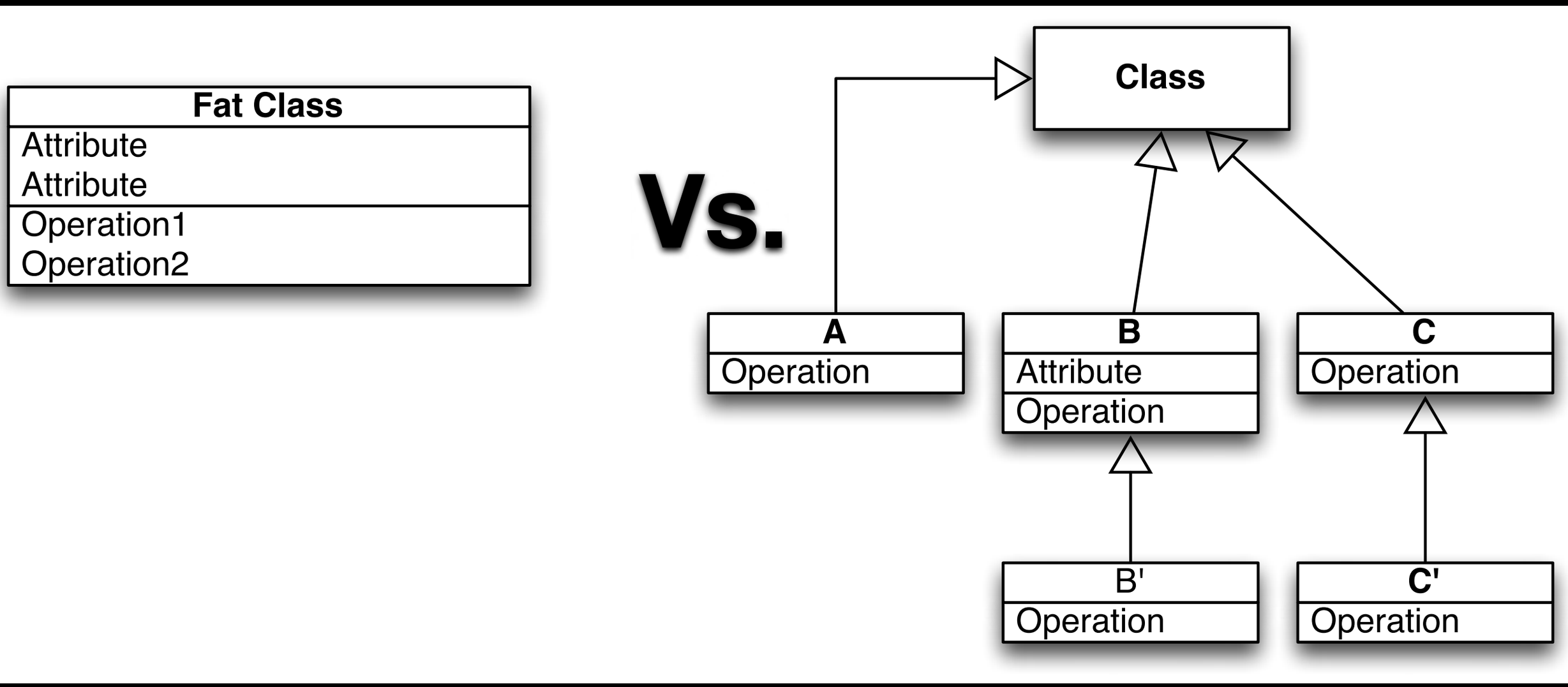
Browse | Senders | Implementors | Version | Source

```
not
  "Negation -- answer true since the receiver is
false."

  ^true
```

Message sends act as case statements

OOP: the Art of Dispatching



Pharo's vision

A powerful engine

to invent (y)our future

An ecosystem where innovation/business bloom

Pharo's Teachers

• Uni. of Buenos Aires • Uni. of Bern • Uni. of Maroua • Uni. of Brussels • Ecole des Mines de Douai • Uni. de Savoie • Ivan Franko Nat. Uni. of Lviv • Czech Technical Uni. • CULS Prague • Uni. of Quilmes • Uni. of La Plata • Northern Michigan Uni. • Uni. Technologica Nacional (UTN) • Uni. Catholic of Argentina • Uni. of Santiago • Uni. Policnica de Catalunya • Uni. de Bretagne Occidentale • Uni. of Tomsk • Uni. of Fernhagen • IT University of Copenhagen • Uni. Cat del Sacro Cuore of Brescia • Uni. Lyon • Uni. Yaounde

Research Groups

Lafhis (AR)
Software Composition Group (CH)
CAR (FR)
RMOD (FR)
Ummisco (IRD)
Reveal (CH)
Lysic (FR)
CEA-List (FR)

Uqbar (AR)
OC (FR)
CCMI-FIT (CZ)
ASERG (BR)
Pleiad (CL)

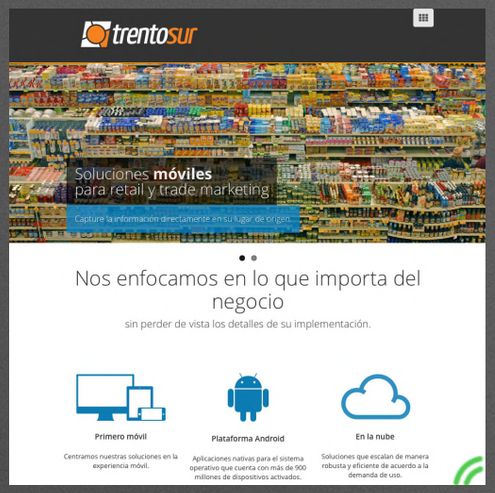
Some Companies

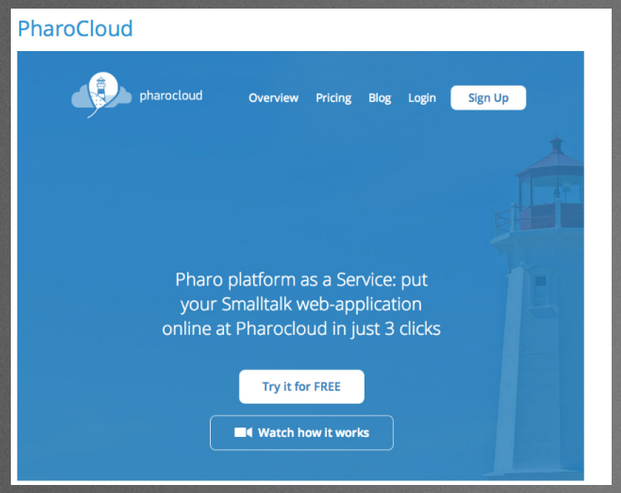
www.2denker.de
www.airflowing.com
www.beta9.be
www.bombardier.com
www.cmsbox.com
www.finworks.biz
seaside.gemstone.com
www.inceptive.be
www.majcon.de
www.mindclue.ch
www.miriamtech.com
www.netstyle.ch
www.panasoft.com
www.pinesoft.co.uk
www.promedmedical.net
www.sharedlogic.ca
www.smallworks.com.ar
www.trantaria.com
www.yesplan.be
www.synectique.eu
www.sorabito.com
www.objectprofile.com
www.pharocloud.com
debrispublishing.com
spesenfuchs.de
norizzk.com

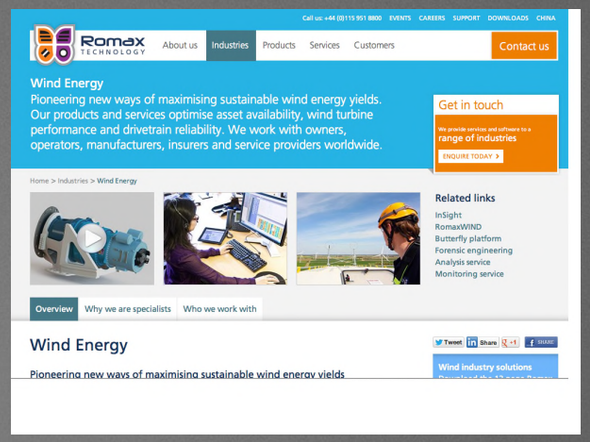
Pharo Web Stack is Gorgeous

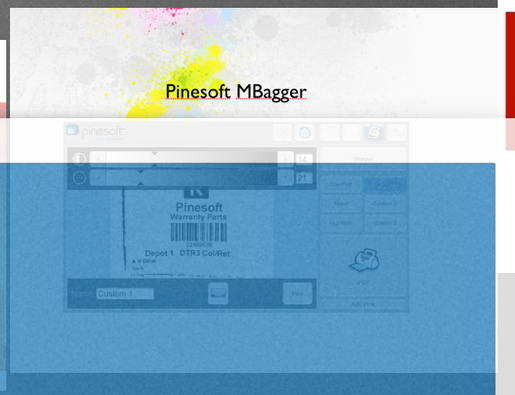
- Seaside (component, Javascript, REST)
- Zinc (HTTP, HTTPS, REST)
- Magritte Metamodelling (no form)
- Protocols/Encoding: Oauth, JSON, STON,…
- Database: noSql, mongoDB, riak, relational databases

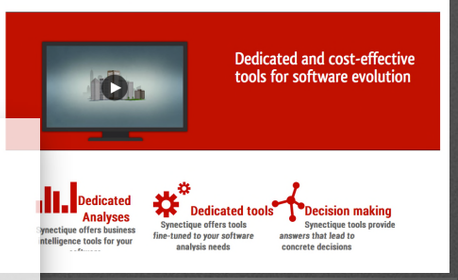
Some Success Stories

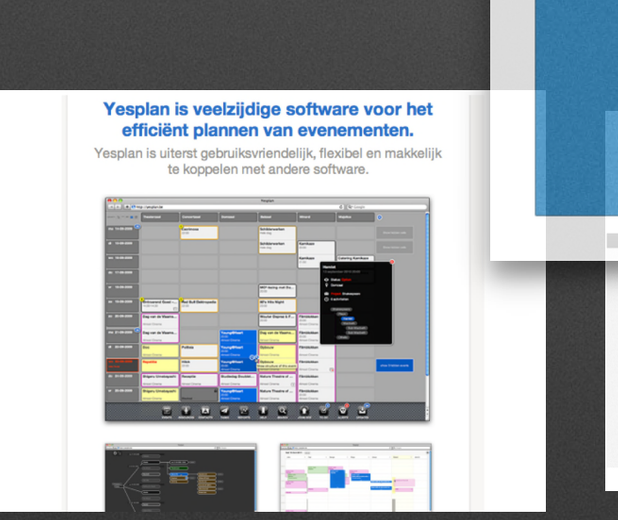


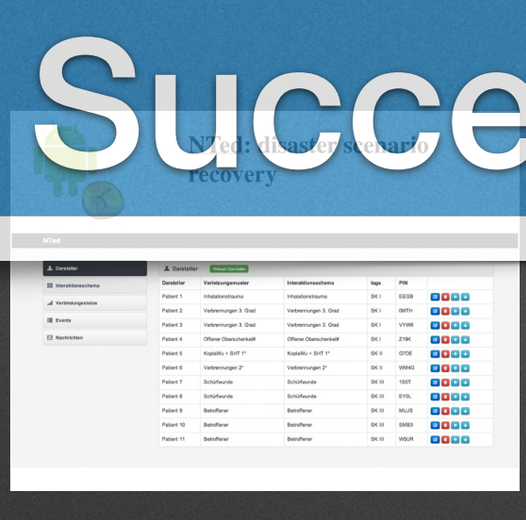


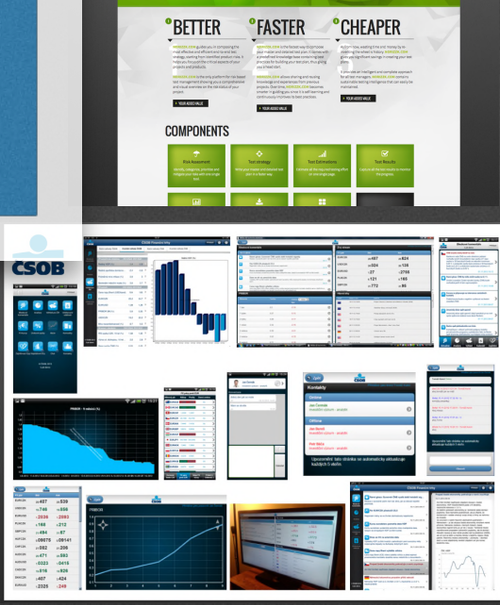


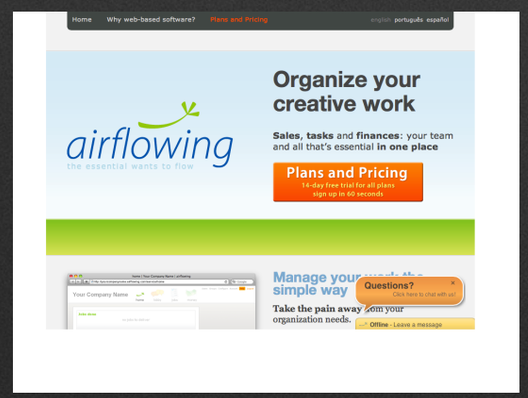


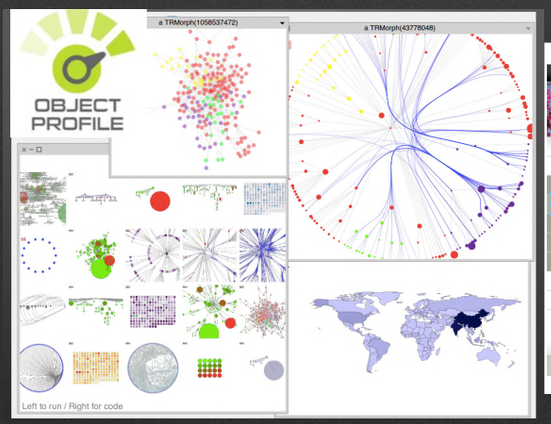


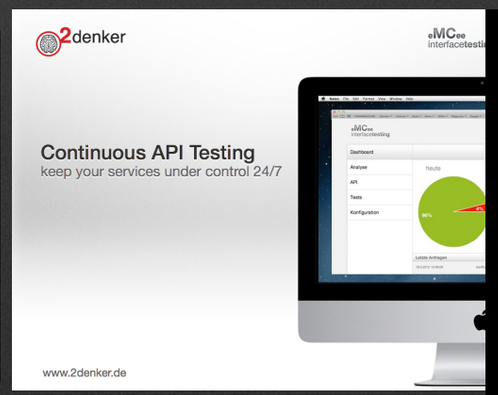


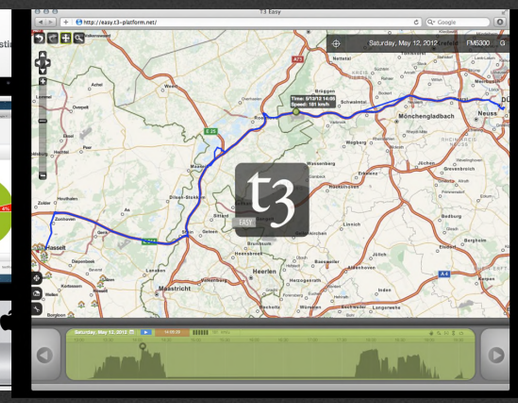












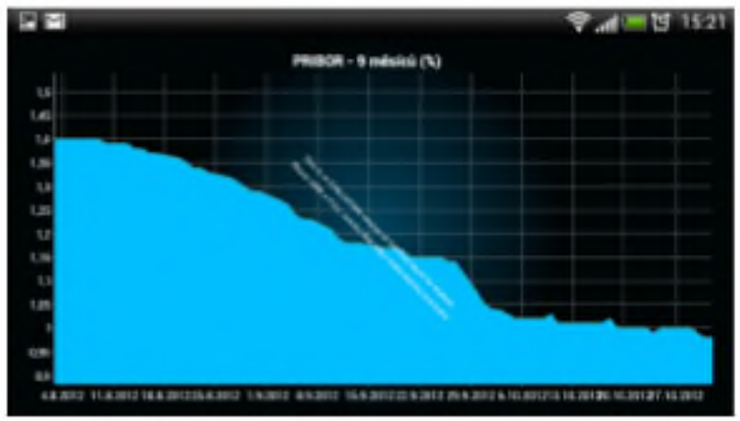
iBizLog - http://www.ibizlog.com

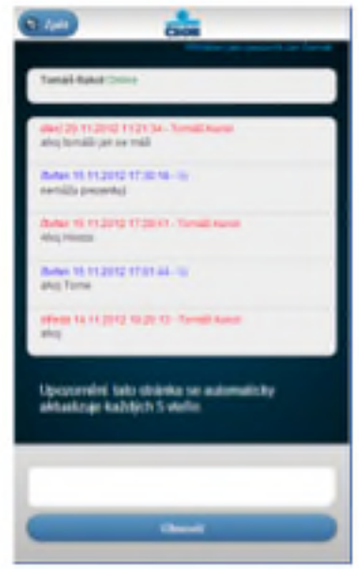
Full meta shop developed in 5 months one person

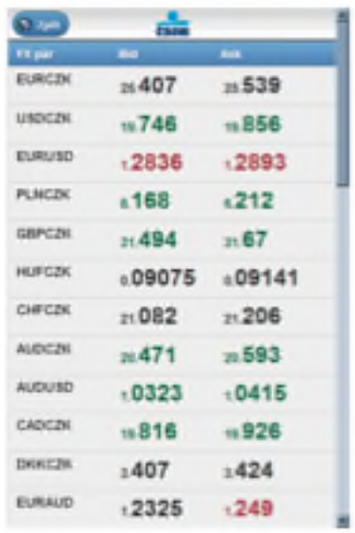
A product by Smallworks

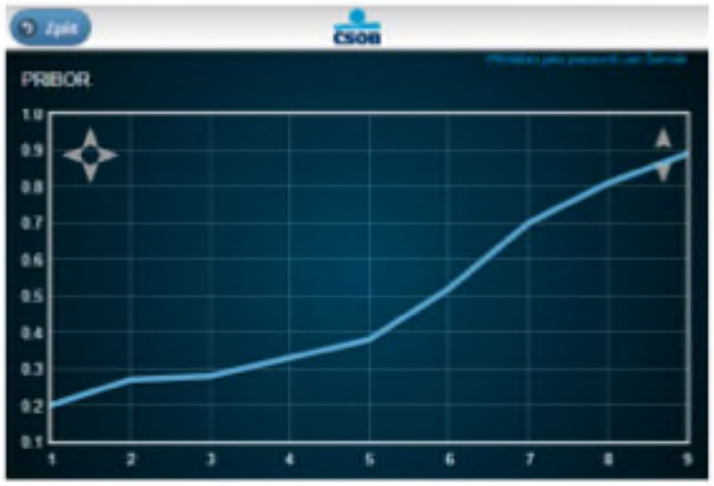
ČSOB

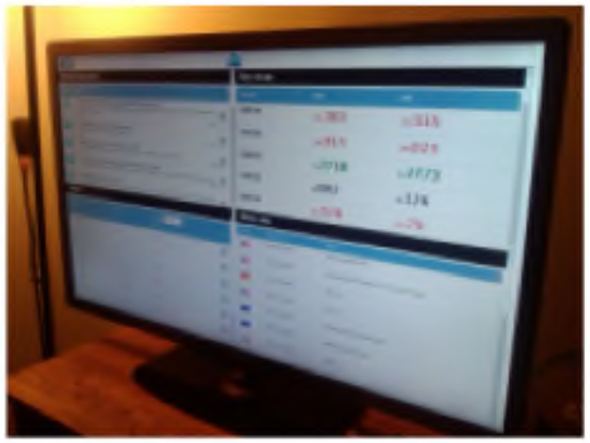
Developed by one developer in three months (Java team estimated 2 years)
Sold to another bank











Google play

drgeo

Apps

Search

Android Apps

All prices

My apps

Shop

Games

Editors' Choice

My wishlist

Redeem

Apps

Dr. Geo free

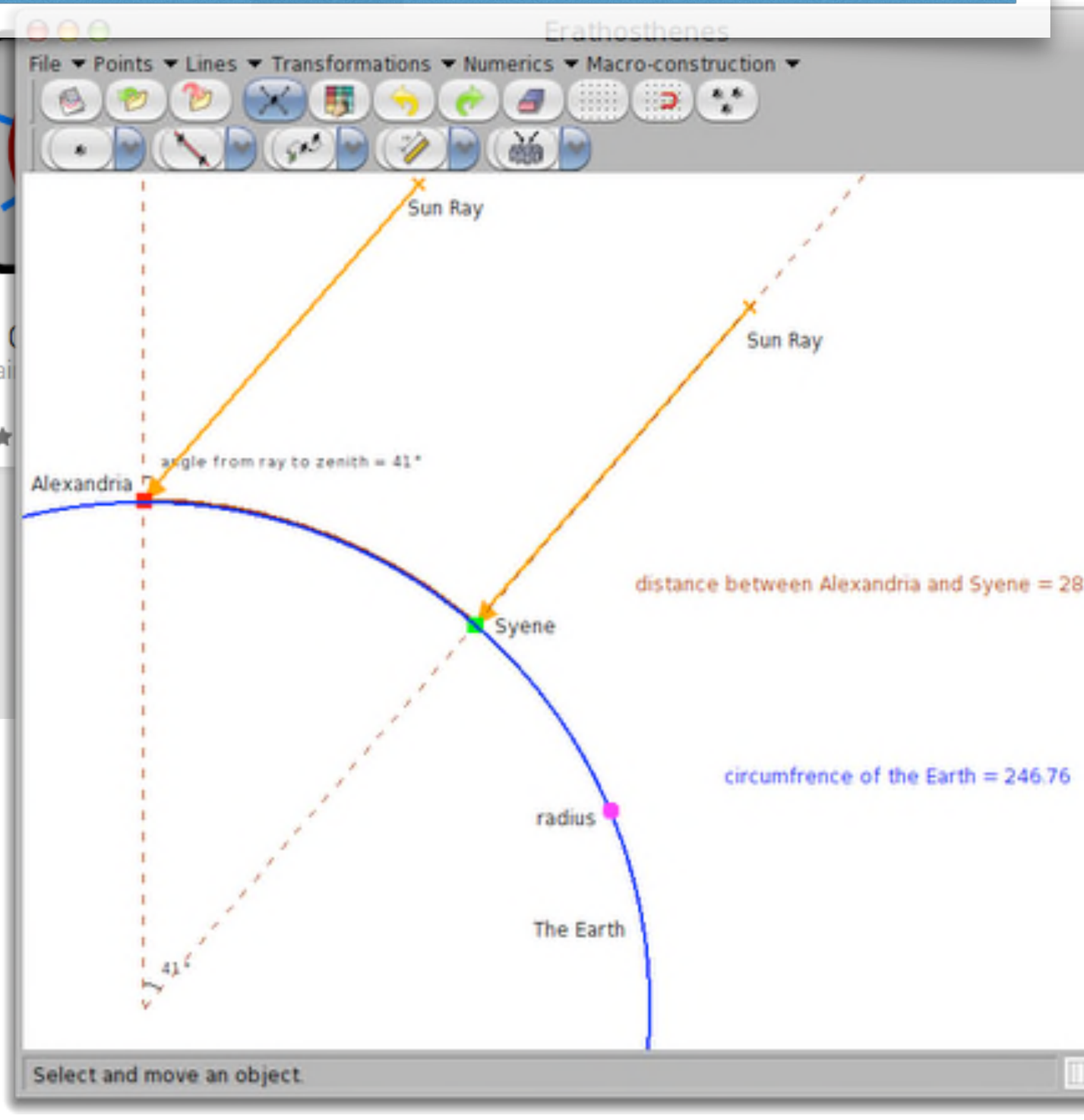
Hilaire Fernandes & Dimi

FREE

Dr.

Hilai

Works on linux, mac, windows, android, OLPC



Pharo is our vehicule

every single day we improve it

Pharo is YOURS

you can learn and help

we feel

responsible

Industrial Consortium

http://consortium.pharo.org

- Promote Pharo
- Sustain Pharo development
- Provide trustable visibility
- Provide support

LABWARE
Results Count

THALES

esug
EUROPEAN SMALLTALK USER GROUP

Lam RESEARCH

Inria
INVENTEURS DU MONDE NUMÉRIQUE

projector
software

Lifeware

dp debris publishing

netstyle.ch

HR works

y

GEMTALK SYSTEMS

2denker

Yesplan
Let's make it happen

BetaNine
software engineering

MAD
ENVIRONNEMENT

inspired!

CæsarSystems
Accelerate Confidently

OBJECT PROFILE

TA MÈRE SCRL
BADASS MOBILE DEVELOPMENT

Systems that make sense
Sensus

Gagnez une Qualité d'Avance

FAST
Fundación Argentina de Smalltalk

MINES Douai
IA
Informatique et Automatique

CZECH TECHNICAL UNIVERSITY IN PRAGUE

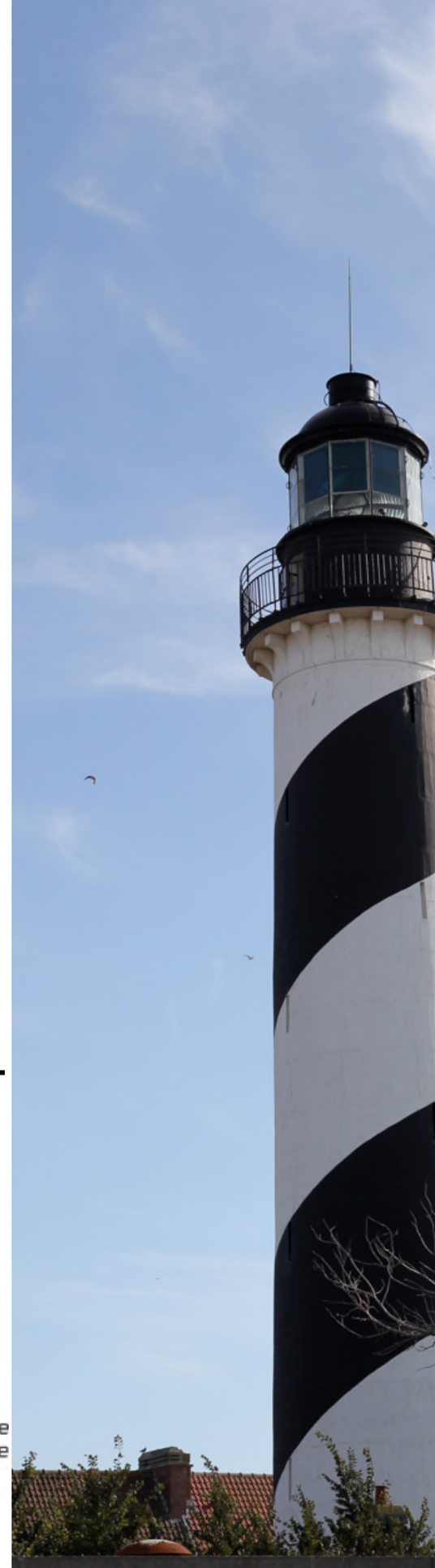
Pleiad

project
uqbar

b
UNIVERSITÄT BERN

RMod

lifl
Laboratoire d'Informatique Fondamentale de Lille



Pharo is an Enabler

“One of the things that drew me to do the Delay refactoring, is simply that I could. That is, I was amazed that I could dig so deep so easily, see a path to improvement and effect change at a fundamental level. … That sense of mastery is seductive.” Ben Coman

Enabler: Turtles all the way

A. Bryant developed Seaside in Pharo's ancestor (he knew ruby, python, scheme, C, objective-C, ...).

Because he could manipulate the stack behind the back of developers. Seaside (http://www.seaside.st) is based on on-demand stack reification.

http://pharo.org

Fun, simple

Excellent for teaching

Pure & elegant

Highly productive

Empowering Tools

Full access

Moose

Moose is a platform for software and data analysis.
It helps programmers craft custom analyses cheaply.

http://moosetechnology.org

A community work

synectique
Inventive Analysis

SCG

RMod

Glamorous Team

Roassal ObjectProfile

Software is

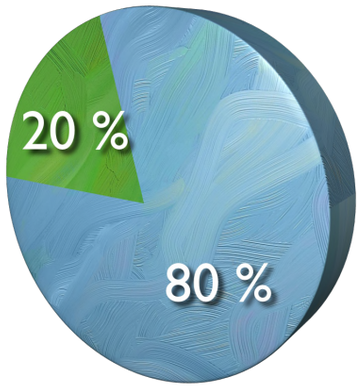
Complex

Software is a living entity...

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

We only maintain useful successful software

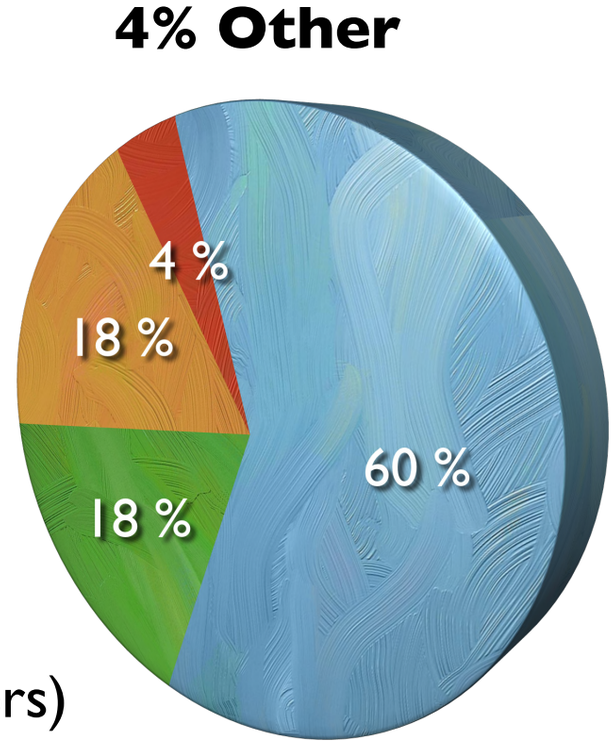
Maintenance is continuous Development



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

18% Adaptive
(new platforms or OS)

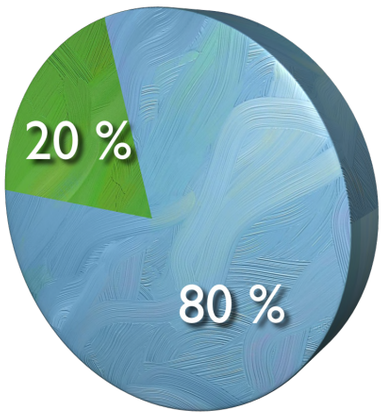
18% Corrective
(fixing reported errors)

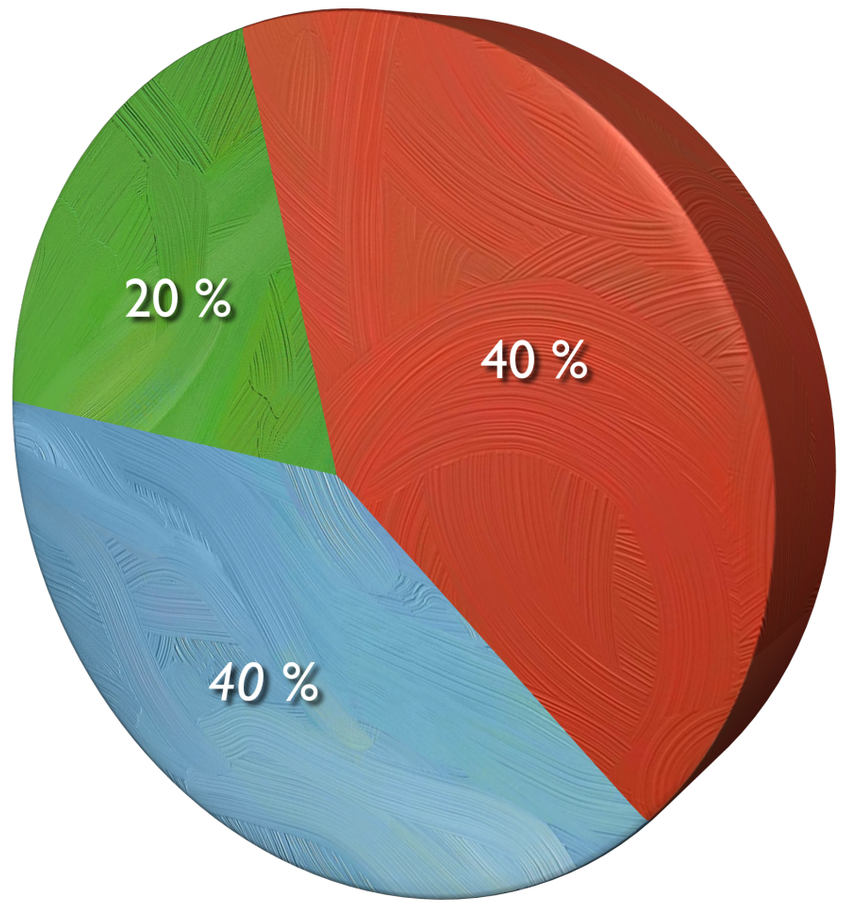


60% Perfective
(new functionality)

"Maintenance"

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





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

We lose a lot of time with inappropriate and ineffective practices

RMOD

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

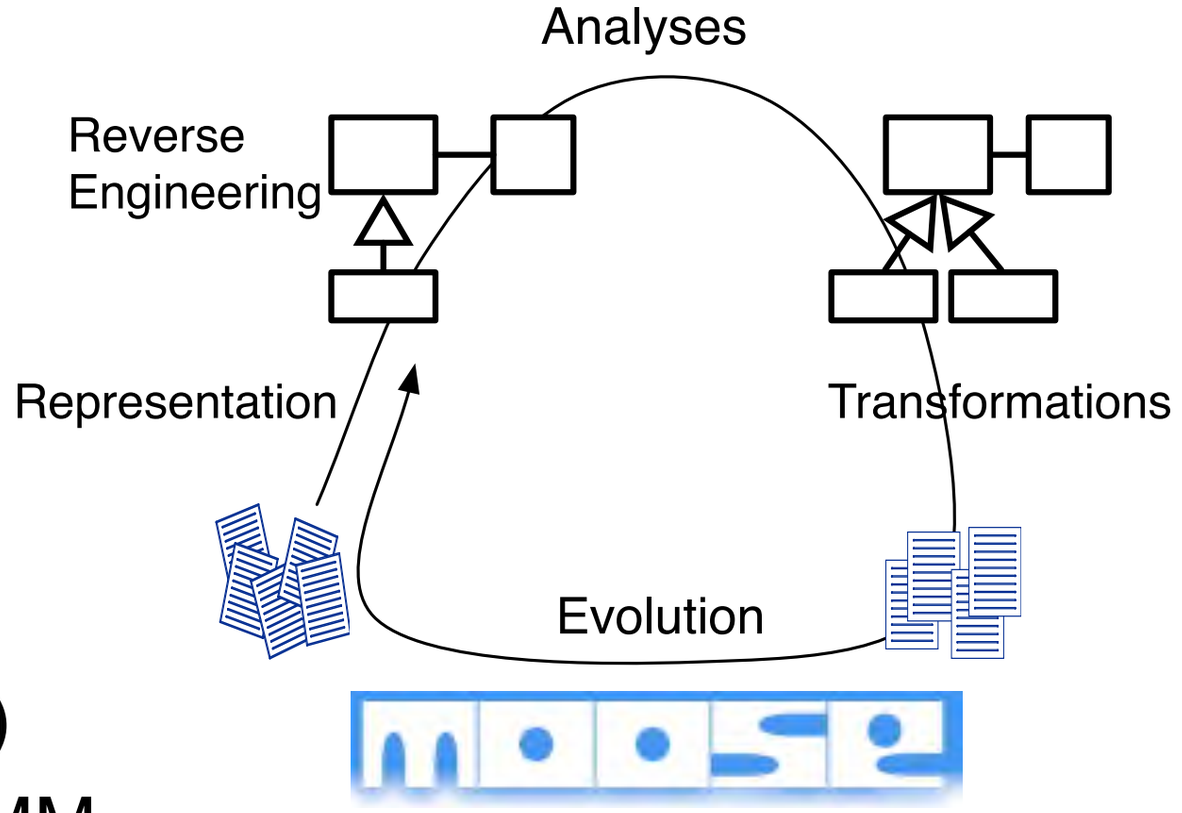
Current focus

- Remodularization analyses
- Quality models (PSA-Airfrance)
- Towards semantic merge
- Old and odd language analyses
- Rule and bug assessment

Collaborations

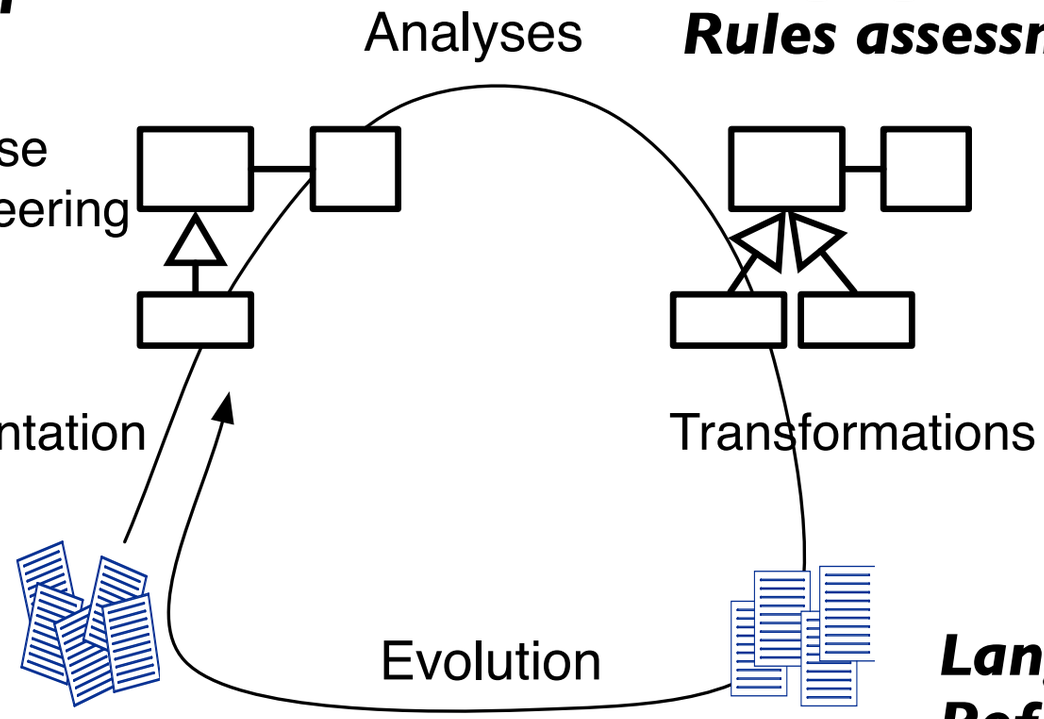
Soft-VUB (Belgium), Pleiad (Chile)

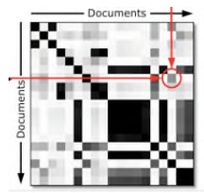
UFMG (Brazil), SCG (Swiss), LIRMM



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

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





Language Independent Meta Model (FAMIX)
An Extensible Reengineering Environment

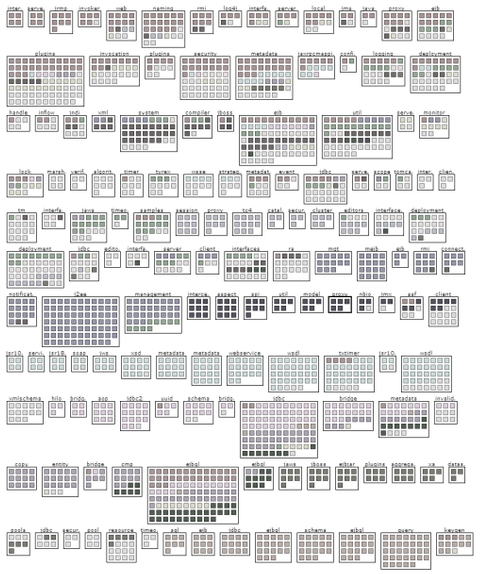
Language Independent Refactorings

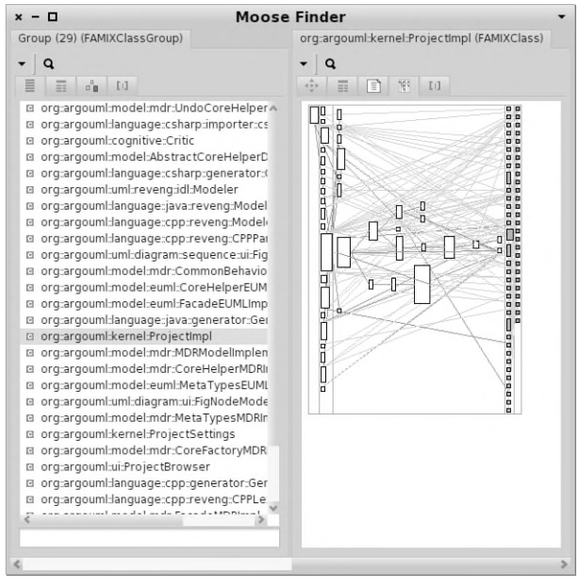
Reengineering Patterns
Version Analyses
Support Evolution
Rules assessment
Dependencies between/inside branches
HISMO metamodel

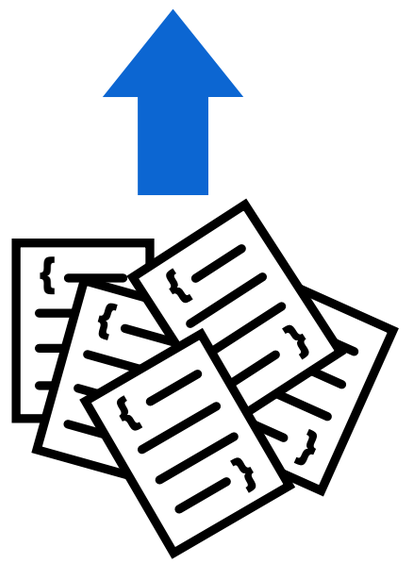
classes select: #isGod

McCabe = 21

LOC = 753,000



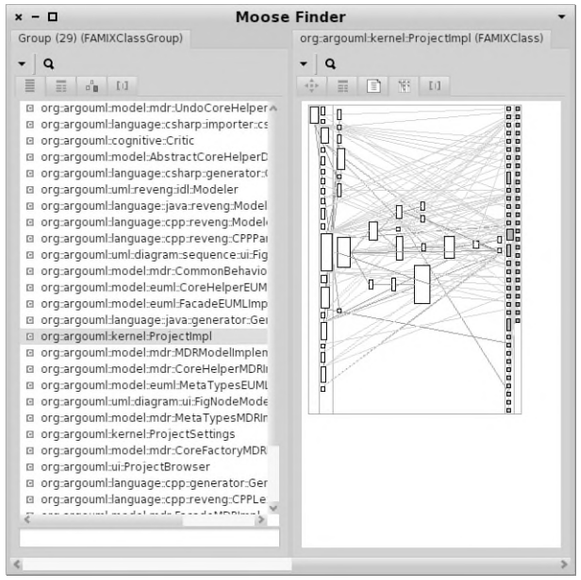


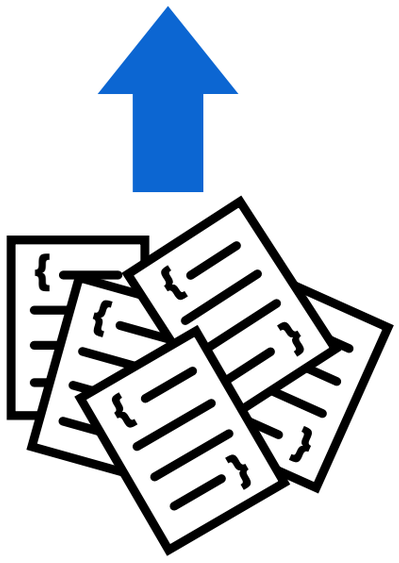


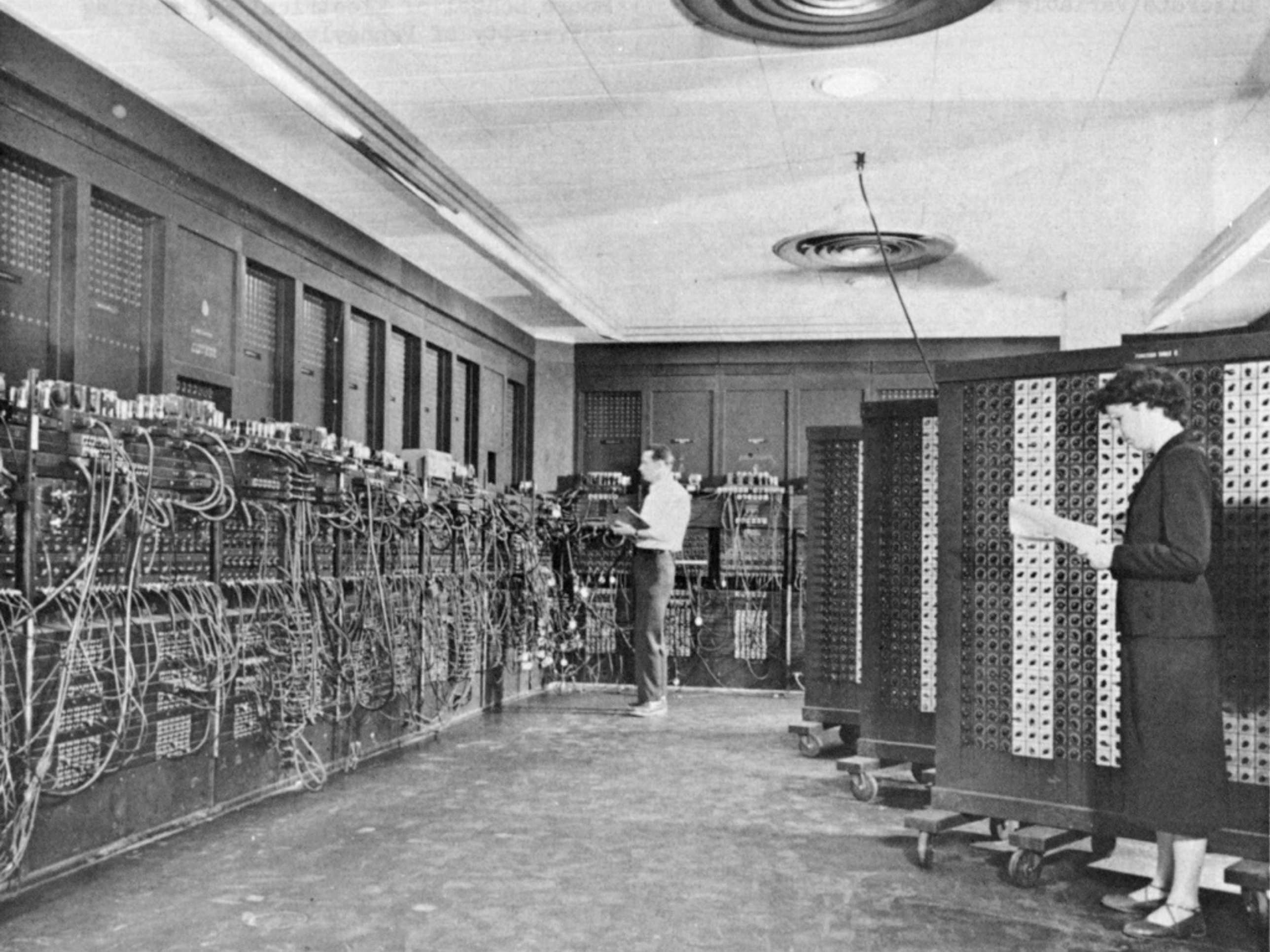
classes select: #isGod

McCabe = 21

LOC = 753,000







A picture tells

a thousand words

FAMIXModelRoot

stub:
stub
isPackage
isMethod
isAttribute
isFormalParameter
isClass
makeStub
isImplicitVariable
isFAMIXObject
isStub

FAMIXAbstractObject

comments
sourceAnchor

hasSourceAnchor
hasSourceAnchorNamed:
sourceAnchor
addComment:
sourceAnchor:
comments

bstractNamedEntity

haracterizationColor

haracterization
ngth
tring
:

FAMIXAbstractAssociation

stateClass

FAMIXInvocation

invokes
invokedBy
candidateList
argumentList
receivingVariable

initialize
printOn:
candidates
invokingBehaviour
argumentList:
invokedBy
makeArgumentList
arguments
setInvokes:
addCandidate:
candidates:
candidateList:
makeCandidateList
anyCandidate
invokes
clearCandidates
clearArguments
argumentList
receivingVariable
addArgument:
smalltalkSelector
numberOfArguments
candidateList
invokedCandidates
setReceivingVariable:
setInvokedBy:
isStub
numberOfCandidates

FAMIXInclude

includingFile
includedFile

includingFile
setIncludedFile:
isInclude
includedFile
setIncludingFile:

FAMIXAccess

readAccess
accesses
accessedIn

accessedIn:
printOn:
read
written
accesses
preOptimizeAccessedInName
isWriteAccess
isClassReference
accesses:
readWriteAccess
accessedIn
accessedVariable
preOptimizeAccessesName
displayString
accessingBehaviour
isReadAccess
setReadWriteAccess:
isAccess
isStub

FAMIXInheritanceDefinition

superclass
subclass
accessControlQualifier
index

initialize
printOn:
preOptimizeSuperclassName
accessControlQualifier
defaultAccessControlQualifier
setSubclass:
protectedSuperclassName
setSuperclass:
superclass
index
subclass
protectedSubclassName
displayString
subClass
isInheritanceDefinition
historicalUniqueName
preOptimizeSubclassName
accessControlQualifier:
superClass
index:
isStub

AMIXAbstractBehaviouralEntity

ccessControlQualifier
gnature
eclaredReturnClass
eclaredReturnType
ackagedIn
elongsTo
comingInvocations
tgoingInvocations
ccesses
rmalParameters
calVariables
comingJspInvocations

itialize
rmalParametersGroup
eclaredReturnClass
calVariables
eclaredReturnType:
eOptimizeBelongsToName
Public
ccessControlQualifier
comingInvocations
calVariablesGroup
umberOfAccesses
ddOutgoingInvocation:
umberOfLinesOfCode
nIn
PureAccessor:
Private
ccesses
ackagedInGroup
eclaredReturnClass:
umberOfFormalParameters
AccessControlQualifierKnown
yclomaticComplexity2
ddLocalVariable:
ivateEncoding
umberOfOutgoingInvocations
BelongsToKnown
otectedEncoding
ddIncomingInvocation:
vokingJspPages
nOut
tSignature:
ePublic
ccesses:
earIncomingInvocations
umberOfComments
umberOfConditionals
ublicEncoding
umberOfStatements

FAMIXAbstractScopable

belongsTo
packagedIn
accessedByList

initialize
preOptimizeBelongsToName
invokedClasses
container
accessedByLists:
preOptimizePackagedInName
packagedIn:
addAccessedBy:
invokedNamespaces
moosePrintOn:
setBelongsTo:
accessedByList
packagedIn
accessedByLists
belongsTo
invokedMethods
accessedByList:

FAMIXPackage

definedClasses
extendedClasses
packagedIn

definedClassesDetect:
initialize
size
definedClassesReject:
allDefinedClasses
preOptimizeExtendedClassesNames:
numberOfLinesOfCode
isRootPackage
extendedClassesIsEmpty
addDefinedClass:
allDefinedClassesDo:
extendedClasses
definedClassesCollect:
viewPackageBrowser
preOptimizeExtendedClassesNames
isPackage
definedClassesIsEmpty
definedClassesDetect:ifNone:
parentPackage
definedClassesSelect:
isLeafPackage
definedClasses
tracesReferencingThisPackage
definedClassesDo:
extendedClasses:
definedClassesContains:
definedClassesGroup
isStubPackage
isHomeFor:
preOptimizeDefinedClassesNames:
definedClassesIncludes:
extendedClassesGroup
packagedIn:
extendedClassesInject:into:
definedClassesInject:into:
definedClasses:
protectedPackagedInNames
preOptimizeDefinedClassesNames
historicalUniqueName
packagedIn
definedClassesOccurrencesOf:
childPackages
addExtendedClass:
totalNumberOfLinesOfCode
isStub

FAMIXJSPPage

filedIn
outgoingJspInvocations

outgoingJspInvocations:
filedIn:
outgoingJspInvocations
filedIn
invokesMethodsFromPackageNamed:
numberOfIncomingJspInvocations

FAMIXJSPAttribute

belongsTo
declaredType
declaredClass

declaredClass
declaredClass:
belongsTo:
declaredType:
declaredType
belongsTo

FAMIXClass

isAbstract
interfaceSignatureSet
attributes
extendedInPackages
methods
isInterface
incomingInheritances
outgoingInheritances

initialize
subclassHierarchyGroup
addInheritsAsSubclass:
outgoingInheritances:
invokedClasses
numberOfBrainMethods
viewCollaborationOn:
isInterface
isInterface:
sourceText
incomingInvocations
incomingAccesses
directSubclasses
withSubclassesDo:
convertSymbol

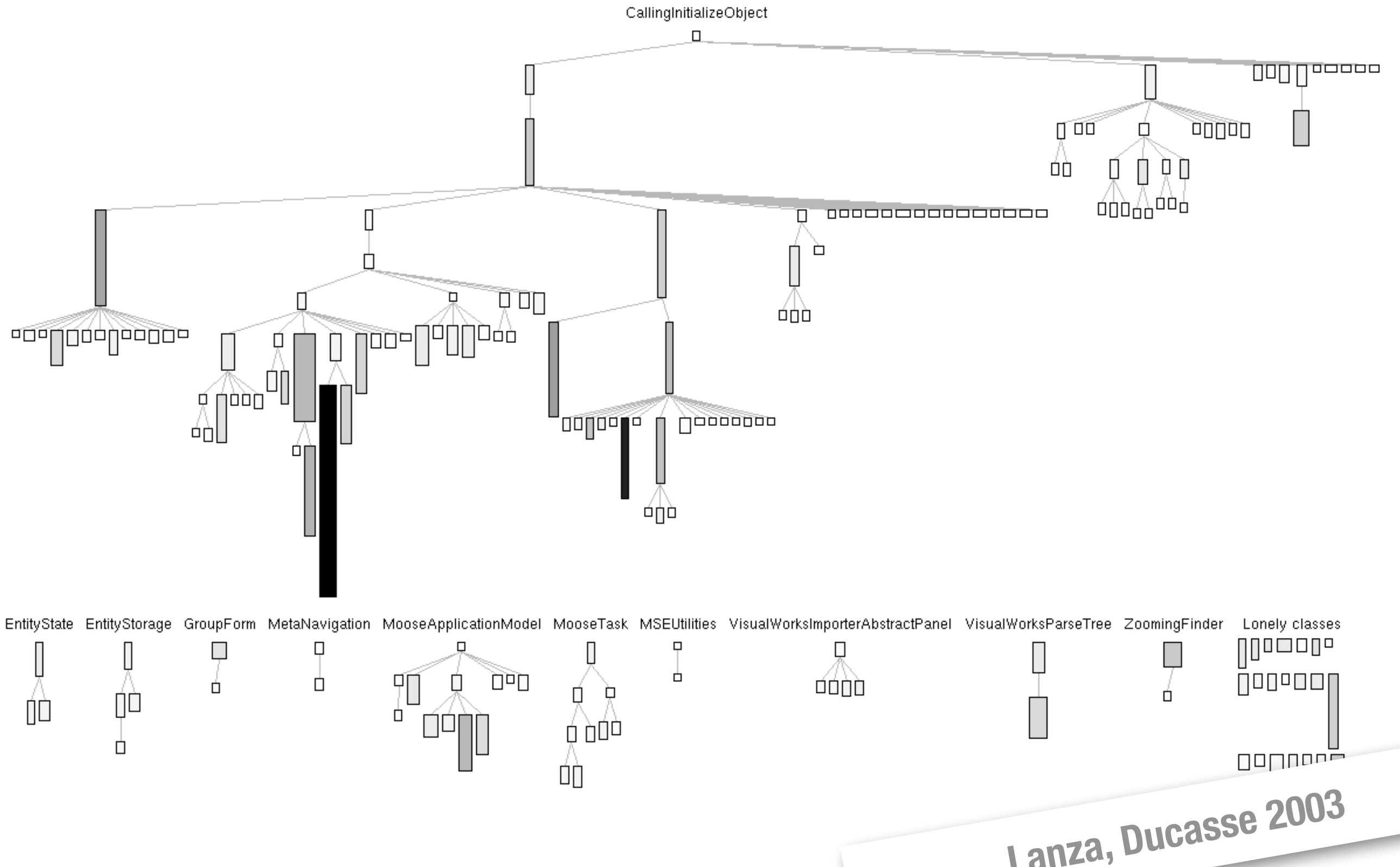
FAMIXNamespace

classes
namespaces
functions

initialize
size
allFunctions
numberOfClasses
recursivelyContainsStubClasses
allClasses
recursiveNamespaces
numberOfLinesOfCode
functions:
functionGroup
numberOfNonInterfacesClasses
addNamespace:
recursiveNumberOfClasses
functions
viewIncomingInvokedMethodsFromNamespaces:on:
recursiveFunctionsGroup
numberOfMethods
addClass:
recursiveClasses
namespaces

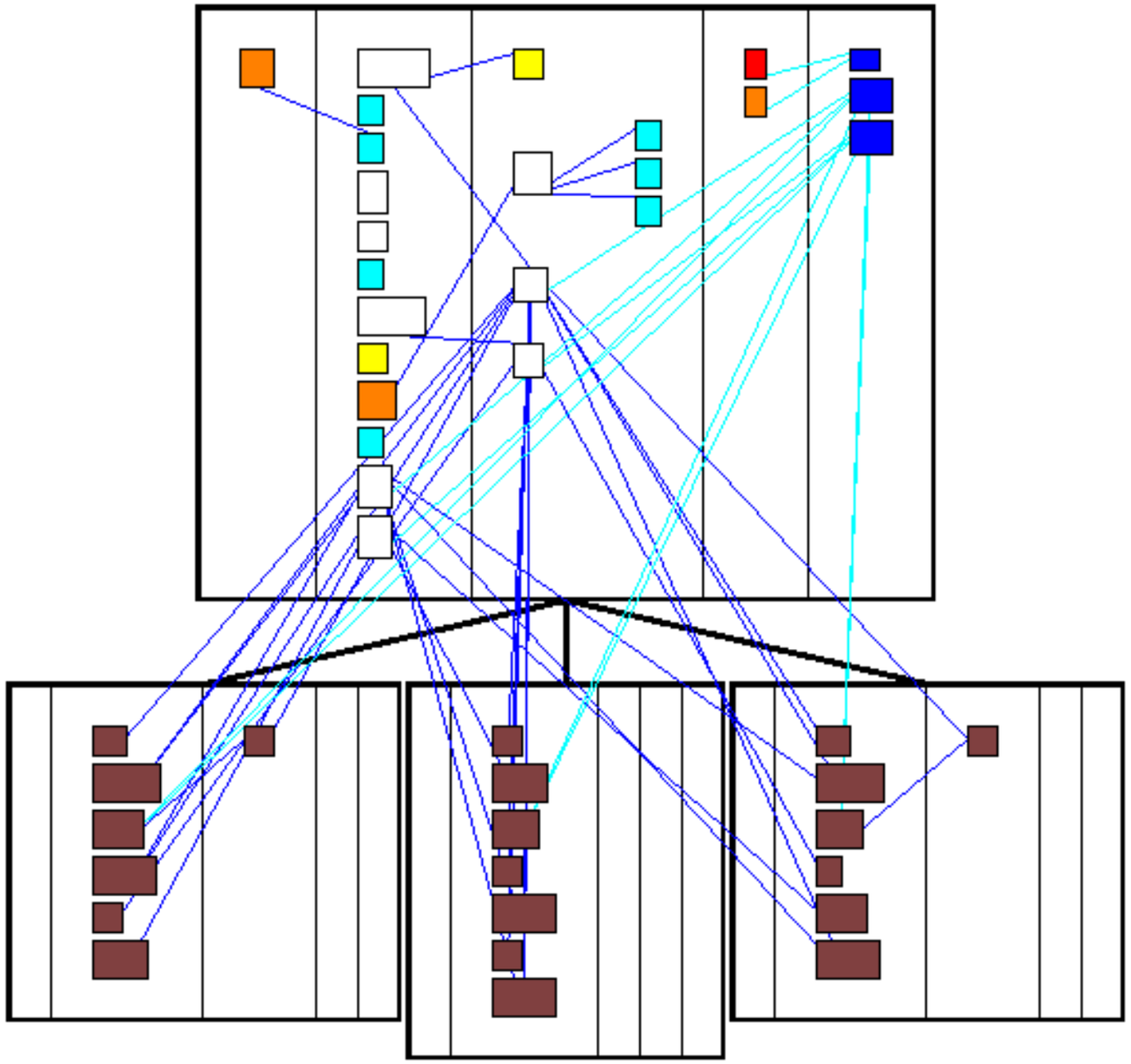
Not all pictures tell a thousand words

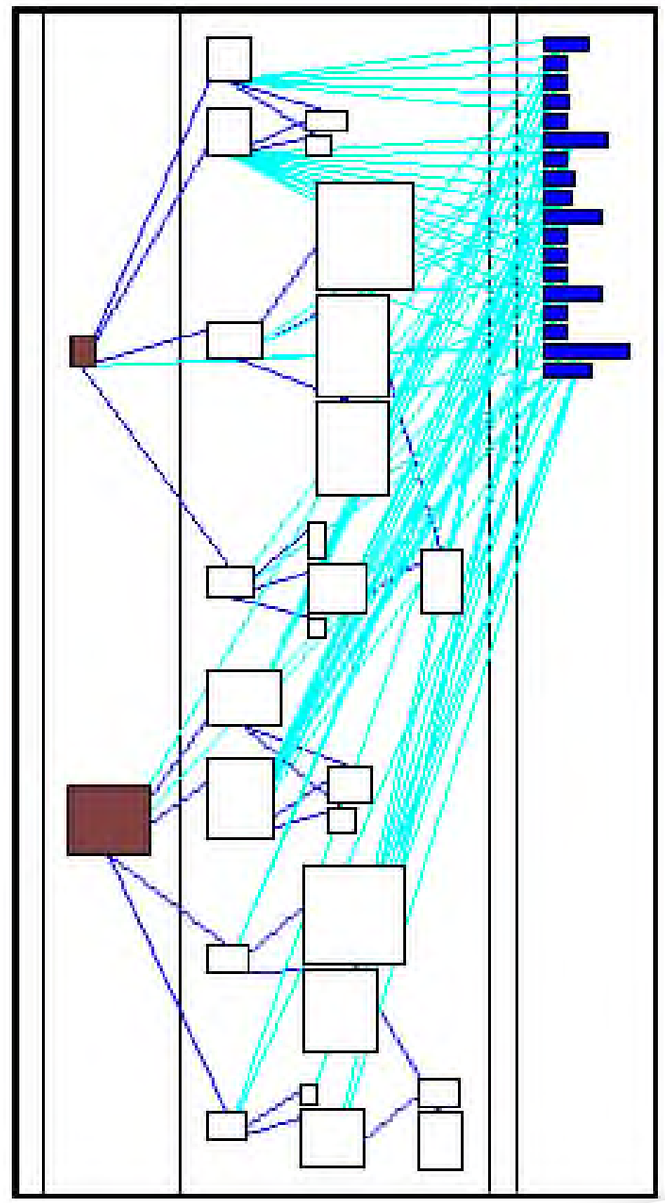
System Complexity shows class hierarchies



Lanza, Ducasse 2003

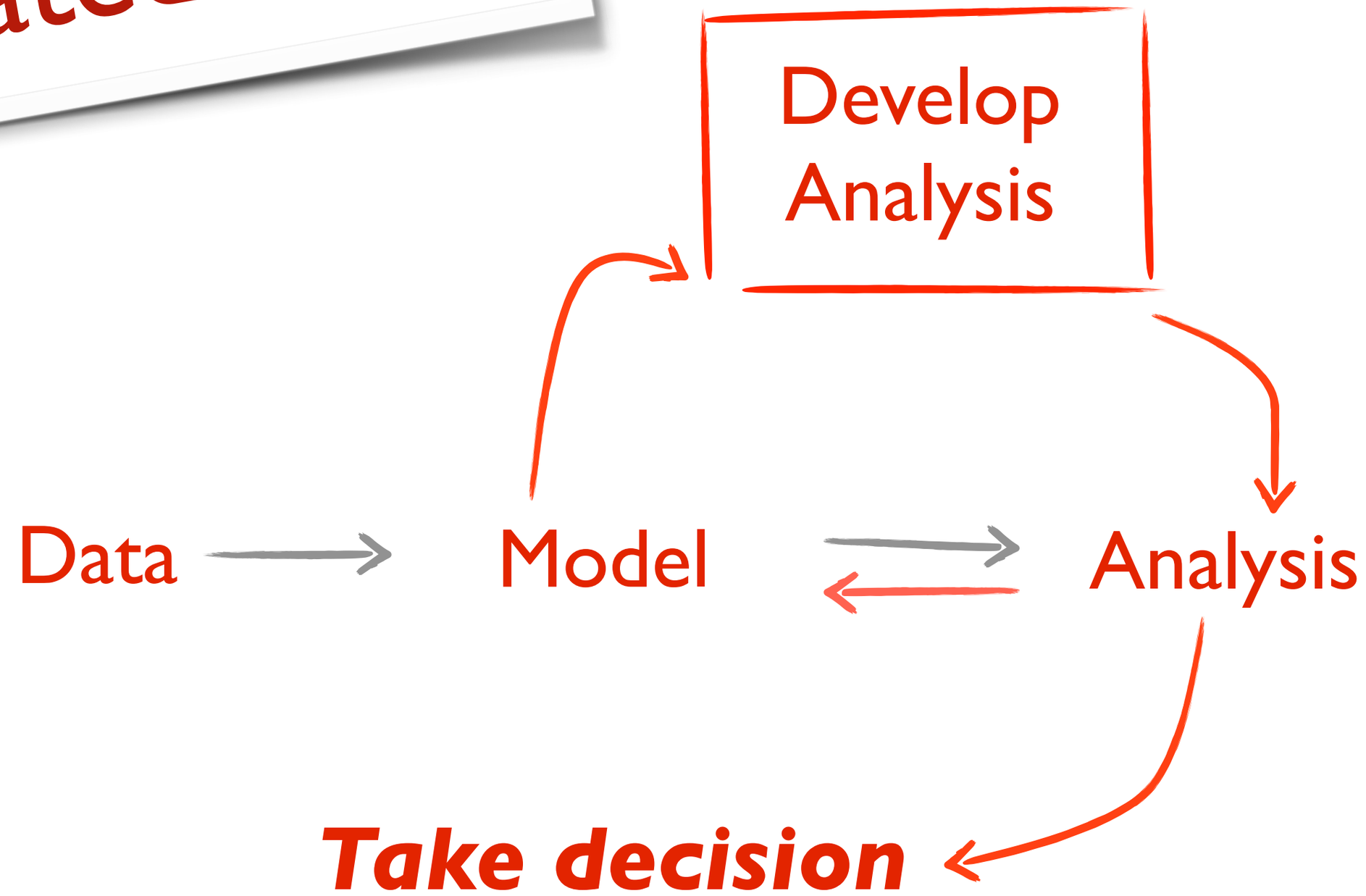
Class Blueprint shows class internals





Ducasse, Lanza 2005

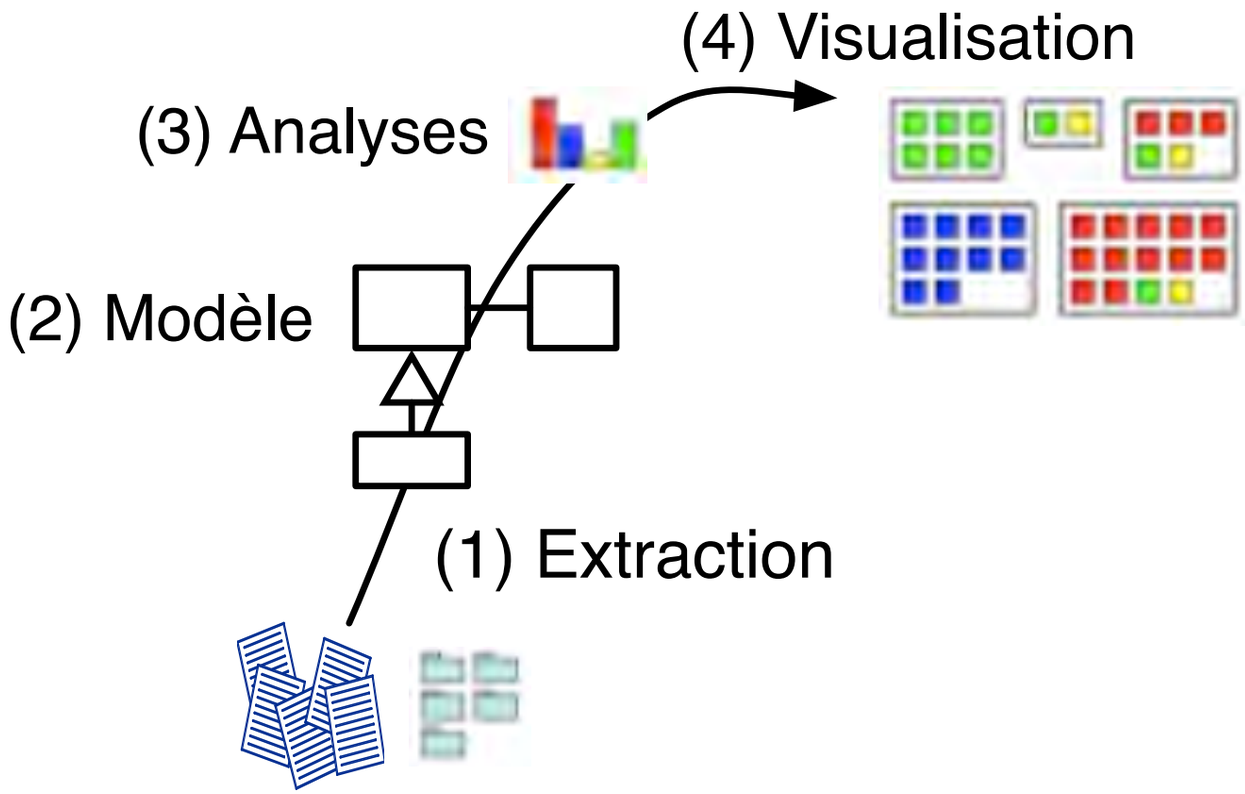
Dedicated Tools



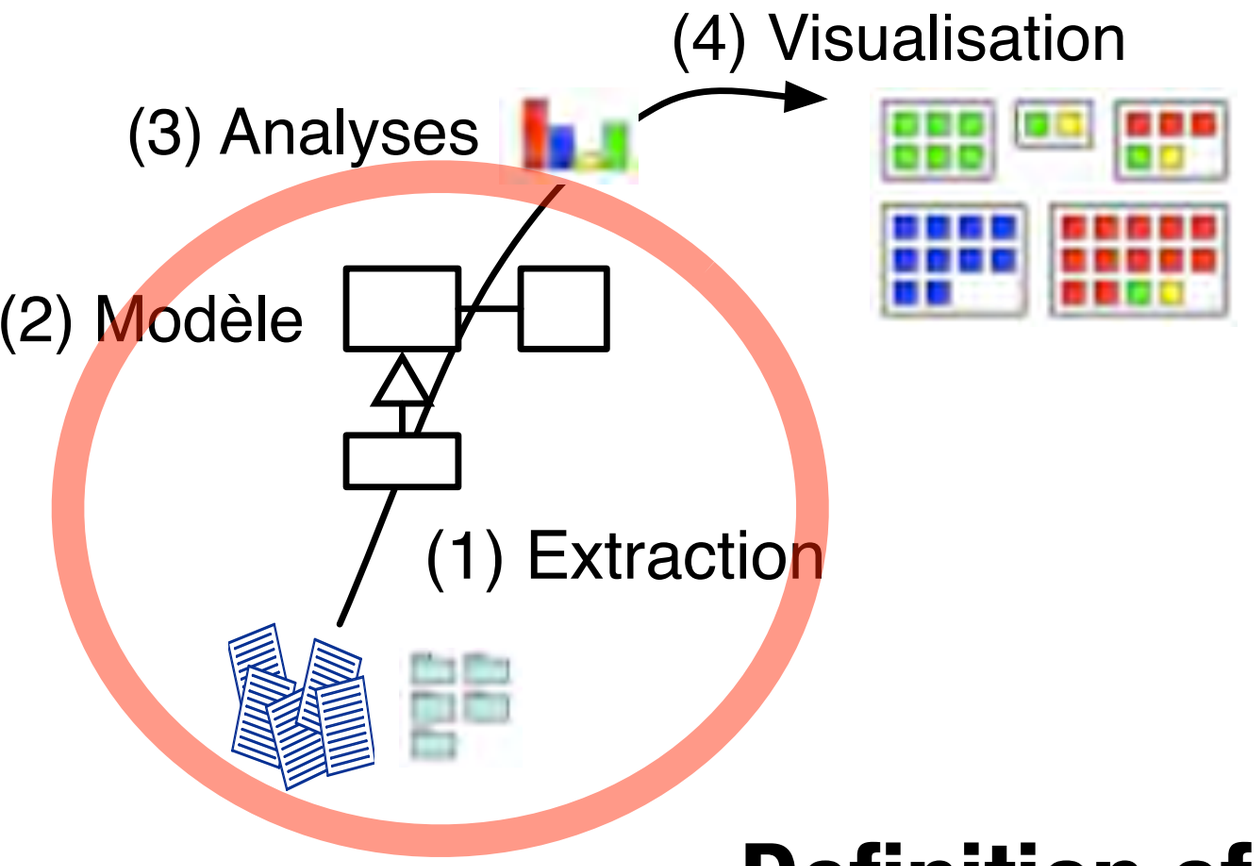
synectique
Inventive Analysis

Analysis should lead to a decision

Example : Who is behind package X ?



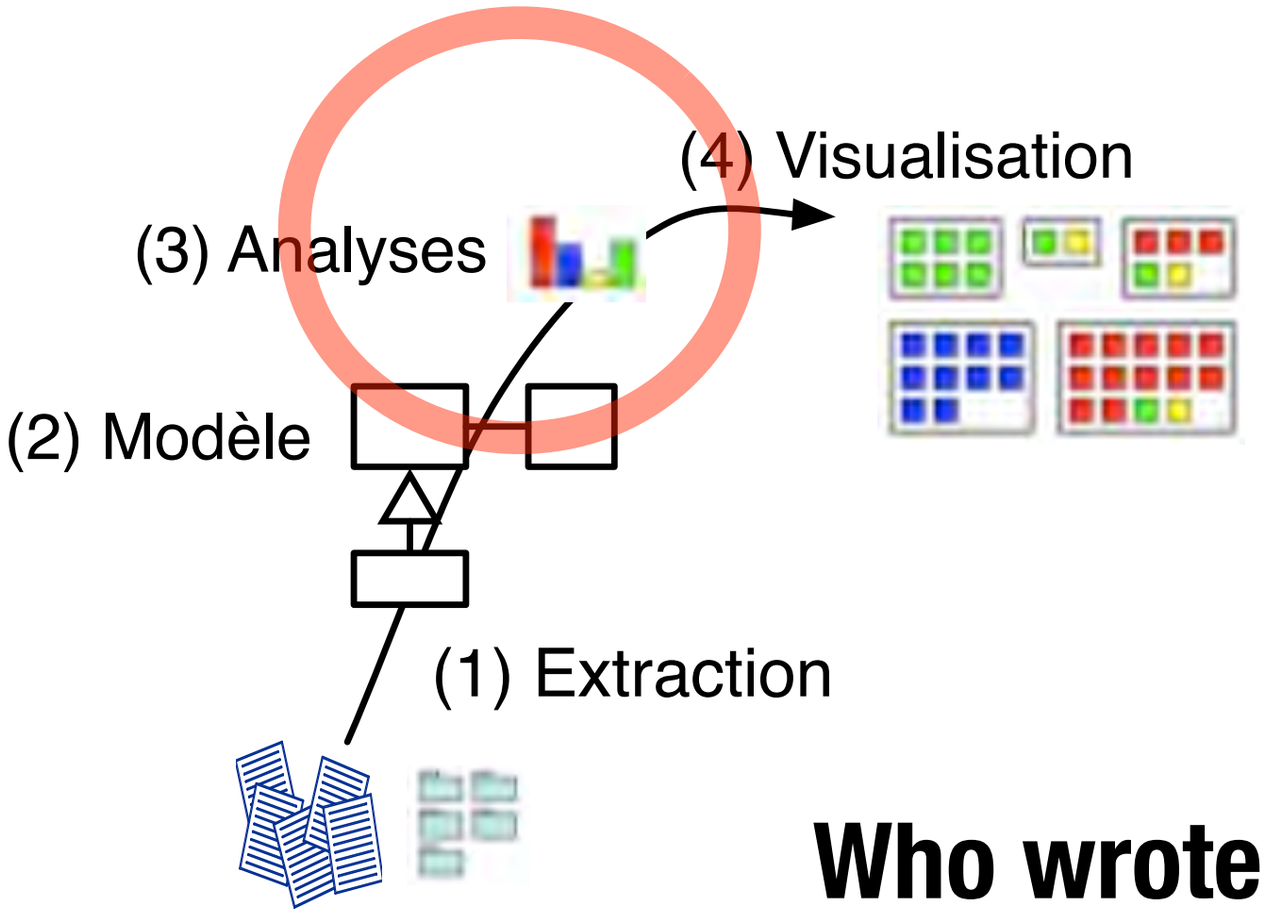
Step 1 - Model Creation/Import



Definition of a model to represent entities

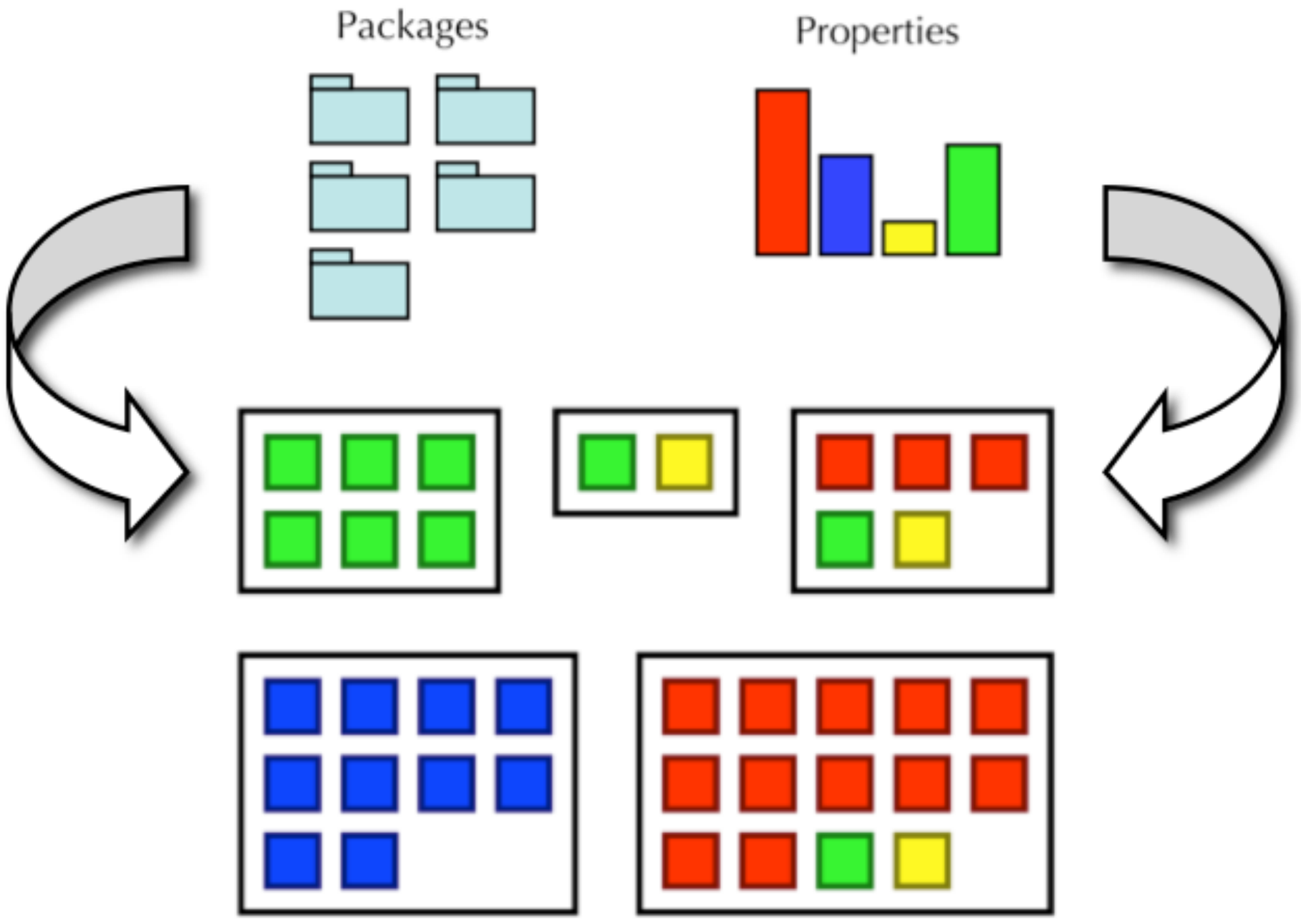
Data Extraction (CVS...)

Step 2 - Analyses



Who wrote how many lines of code?

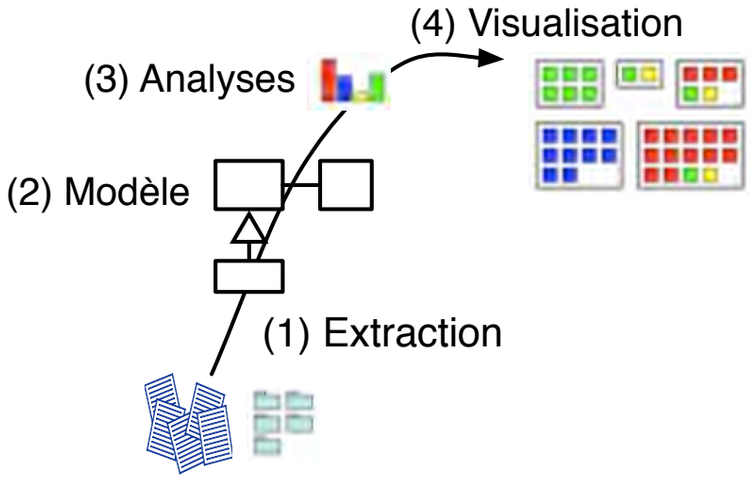
Step : 3 - Creating the Map

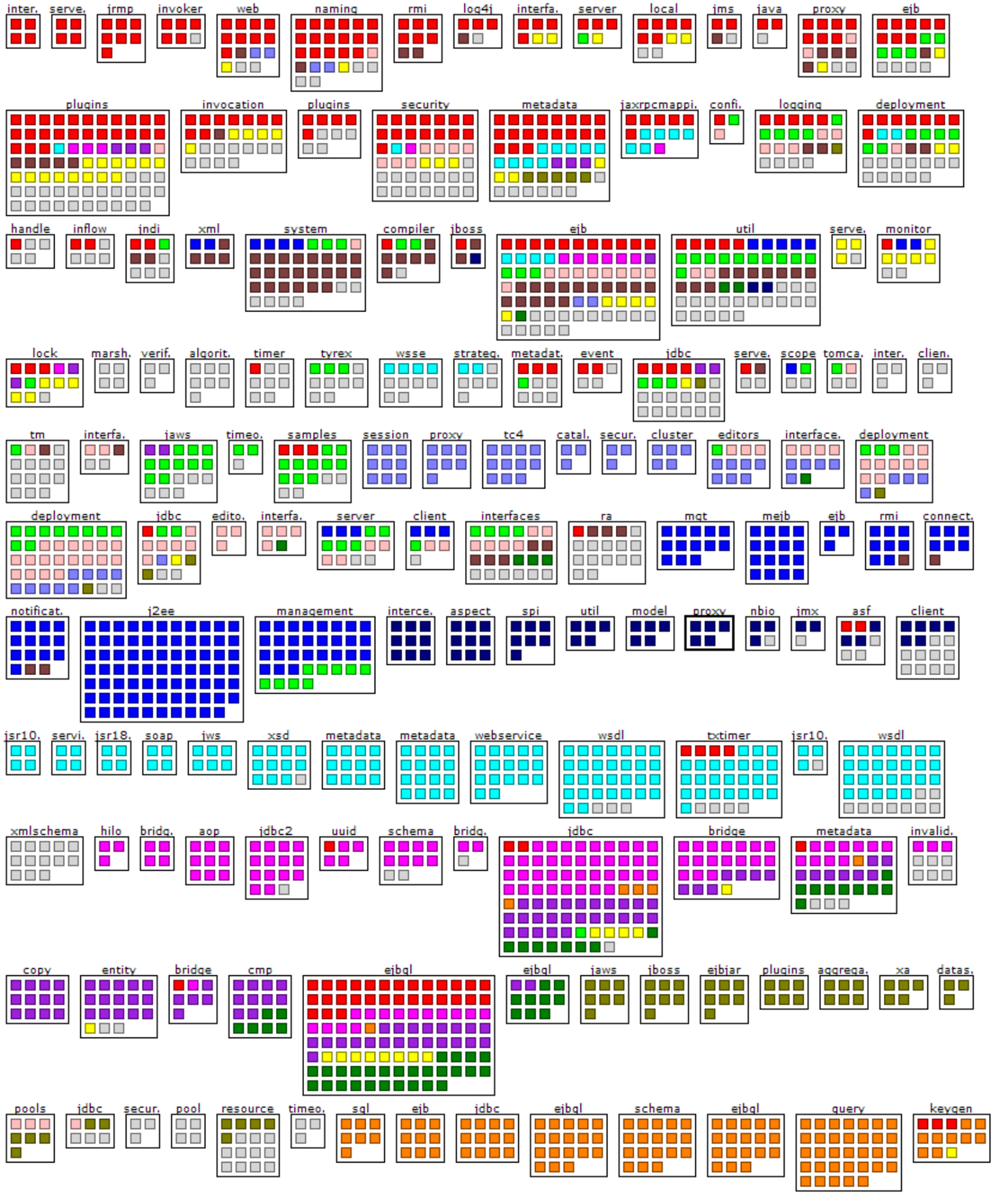


JBoss at a glance

Interactive tool

Data in perspective

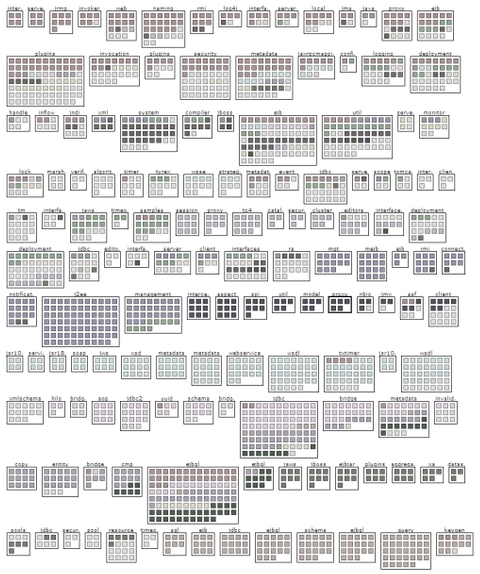


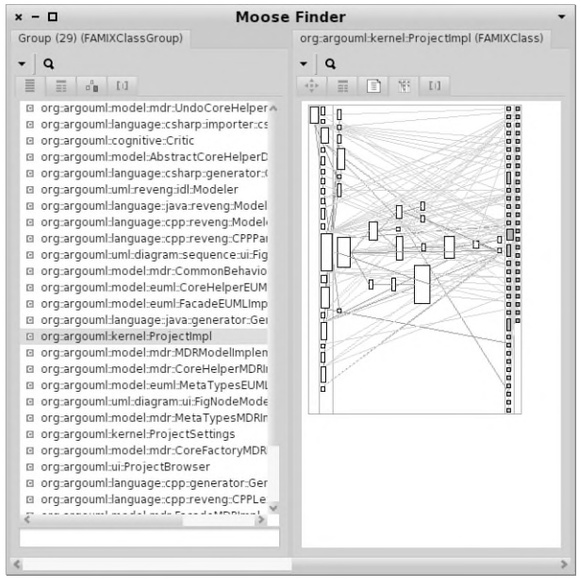


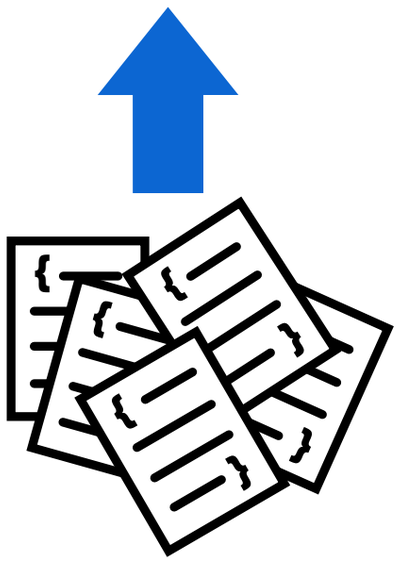
classes select: #isGod

McCabe = 21

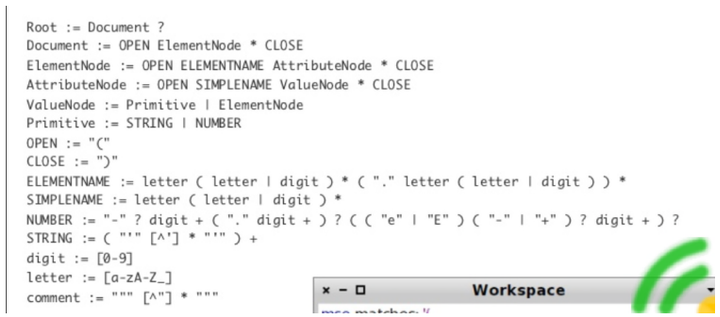
LOC = 753,000

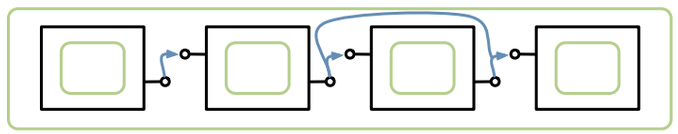






Moose is a tool ecosystem



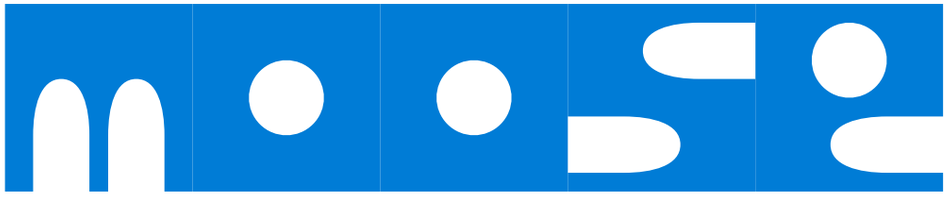


PetitParser

Dynamic Grammars in Smalltalk

glamour

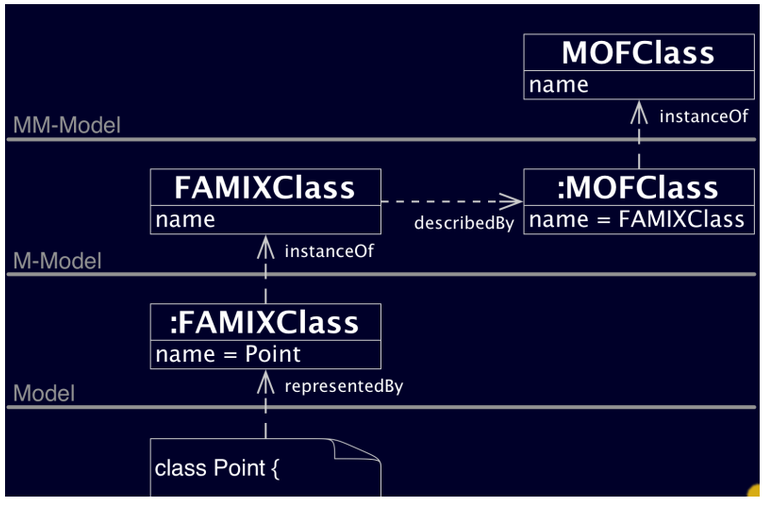
MooseChef



FAME

FAMIX

& other



Mondrian/Roassal scripts graph visualizations

Meyer etal 2006

Thanks
ObjectProfile
and A. Bergel
for Roassal

Playground

Playground

Playground

RTBox

a TRMorph(867958784)

import code | layout | select | color | Clean | remove

Trachel label

The kind of gift you can get from a community :)

A glimpse at RMOD research

(pharo is not our research)

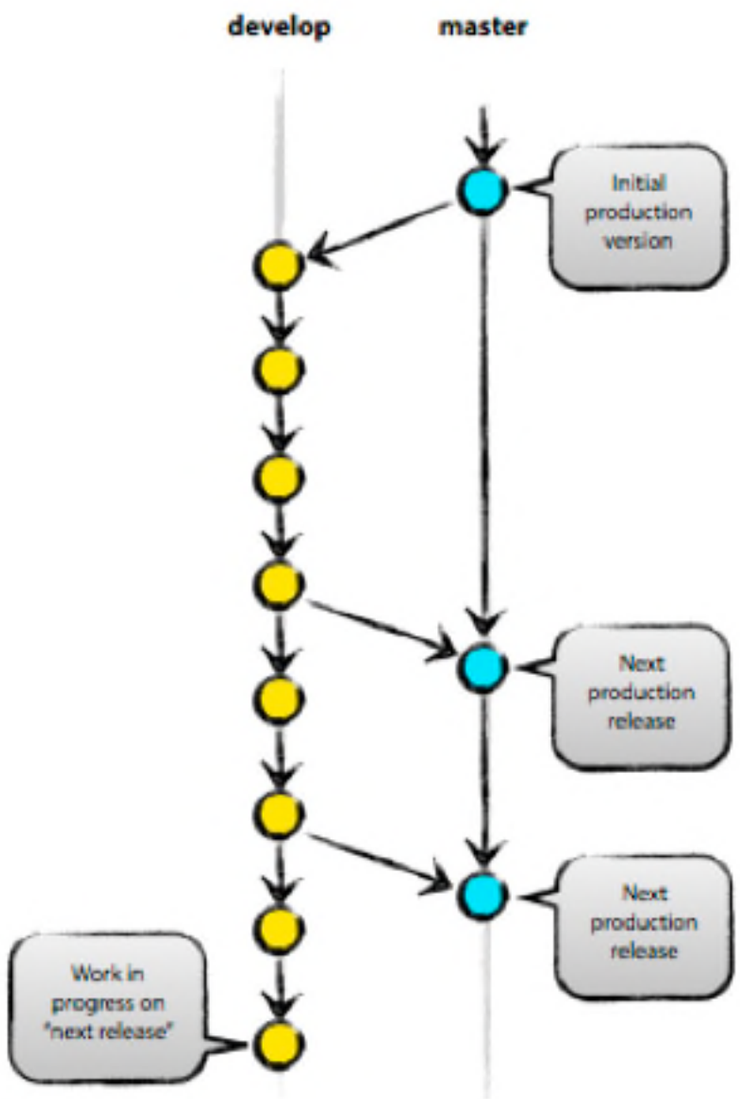
Supporting changes

Supporting merge (PhD of V. Uquillas-Gomez)

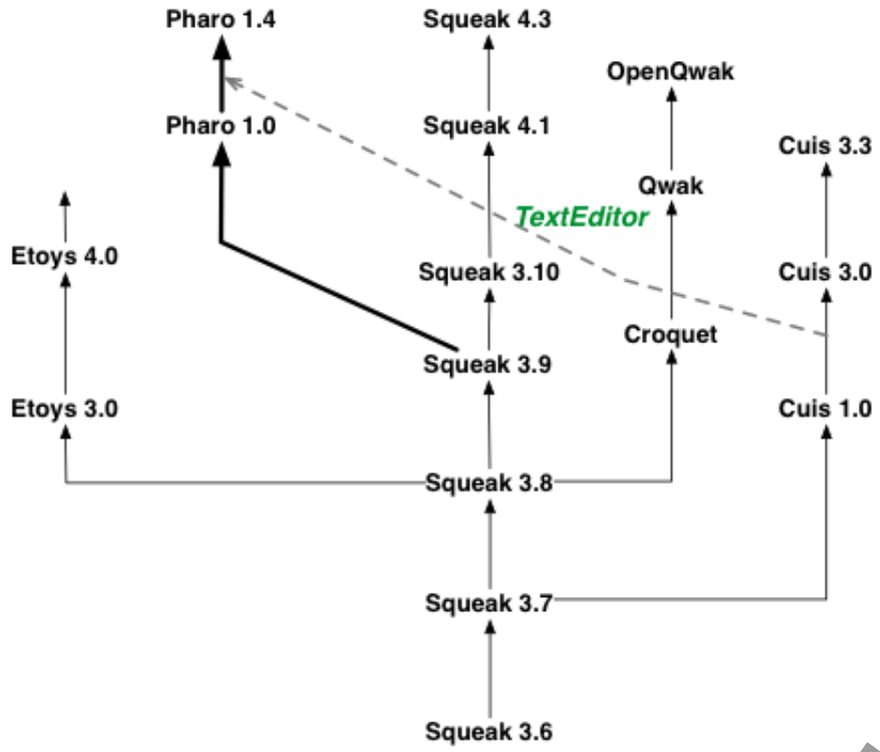
Untangling changes (Ph.D. of M. Dias)

Helping evolving your system (Ph.D. of A. Hora)

How to support merging branches?



Git



Forks

Manual tasks are needed

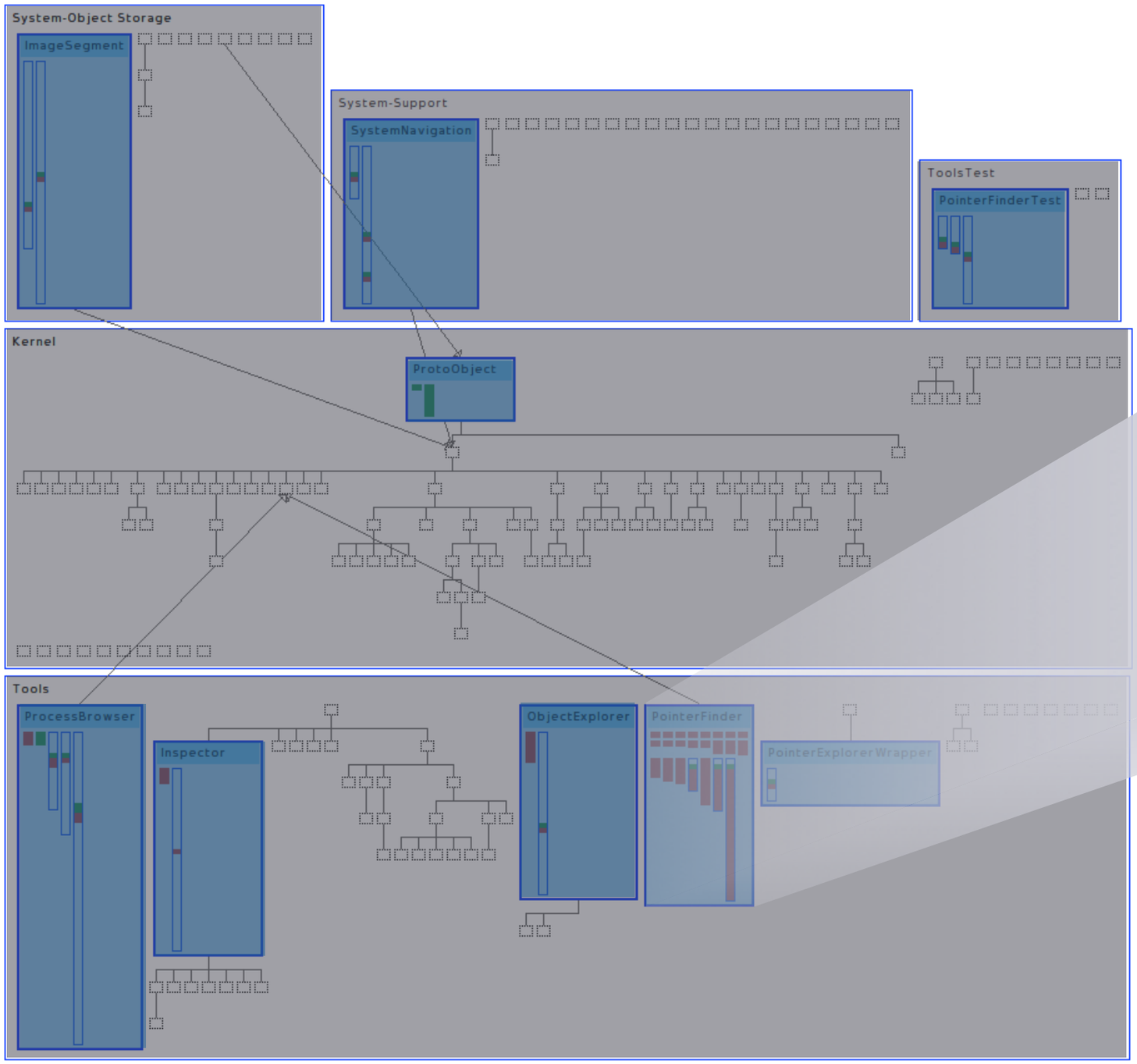
Dependencies between changes

Integrator is not the author of the cha

No

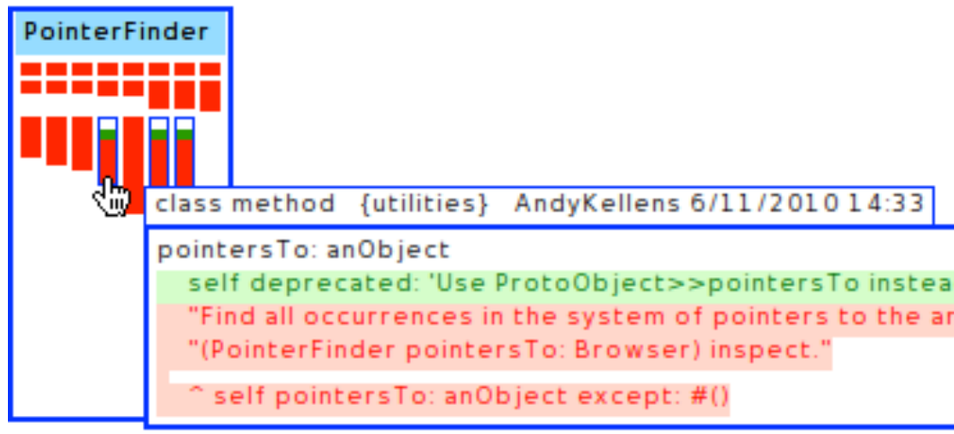
V. Uquillas-Gomez

Torch: Which changes?
Where? Who? What?

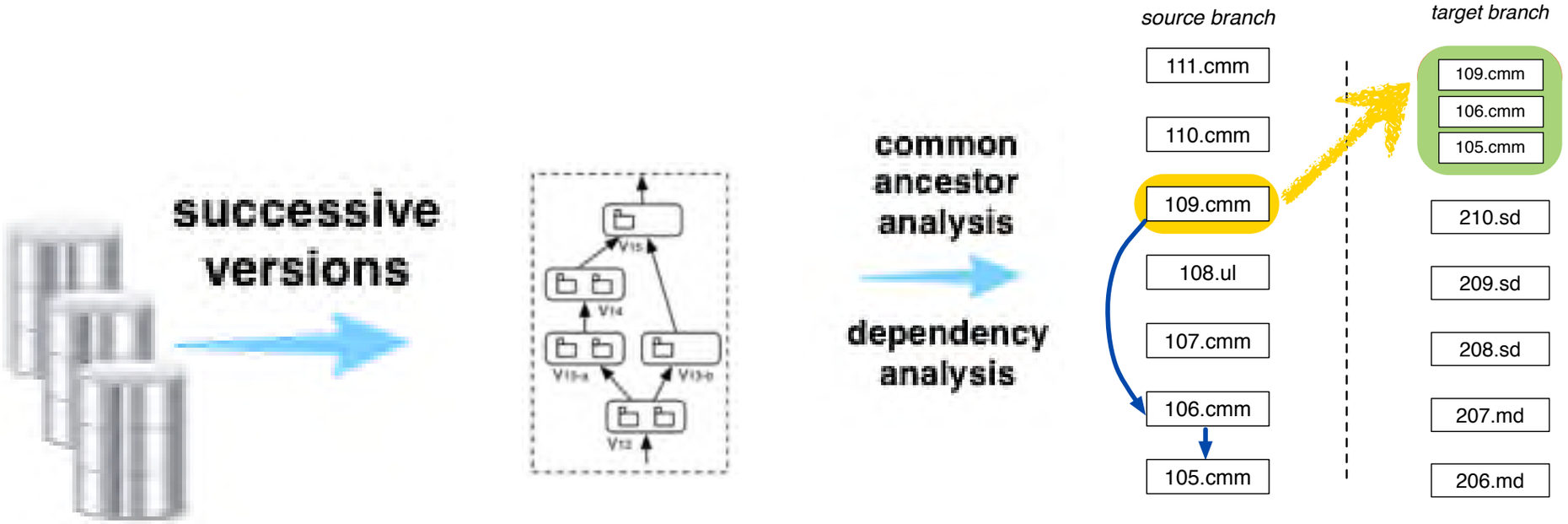


A set of changes, involving:

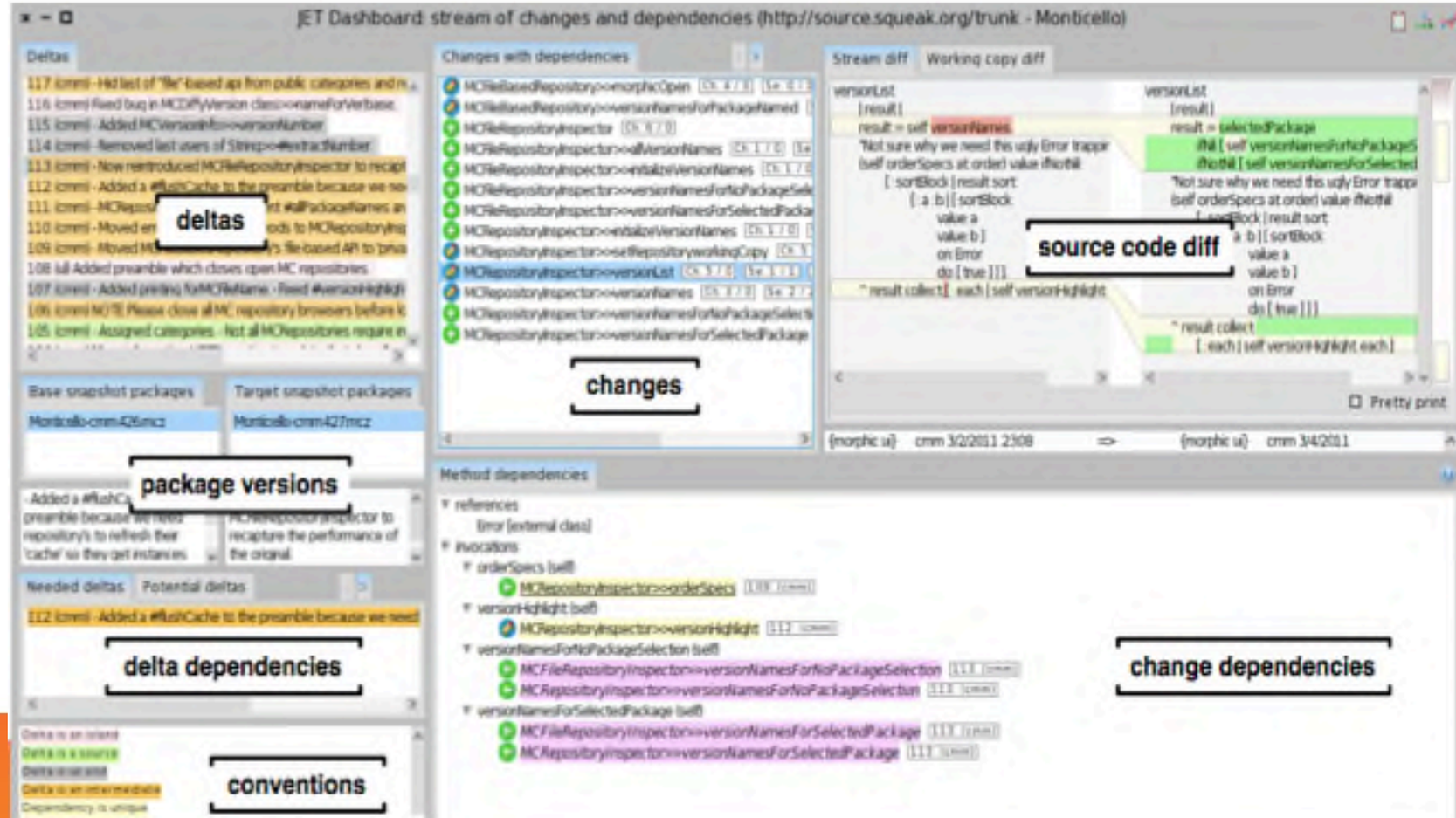
- 5 packages,
- 9 classes,
- ~40 methods



Streams of Changes:
On what other changes does this change depend?



characterization of dependencies and deltas



Untangling changes

Feature #6

Fix Bug

Feature #6

t

CH 1 add method

CH 2 modify method

CH 3 modify method

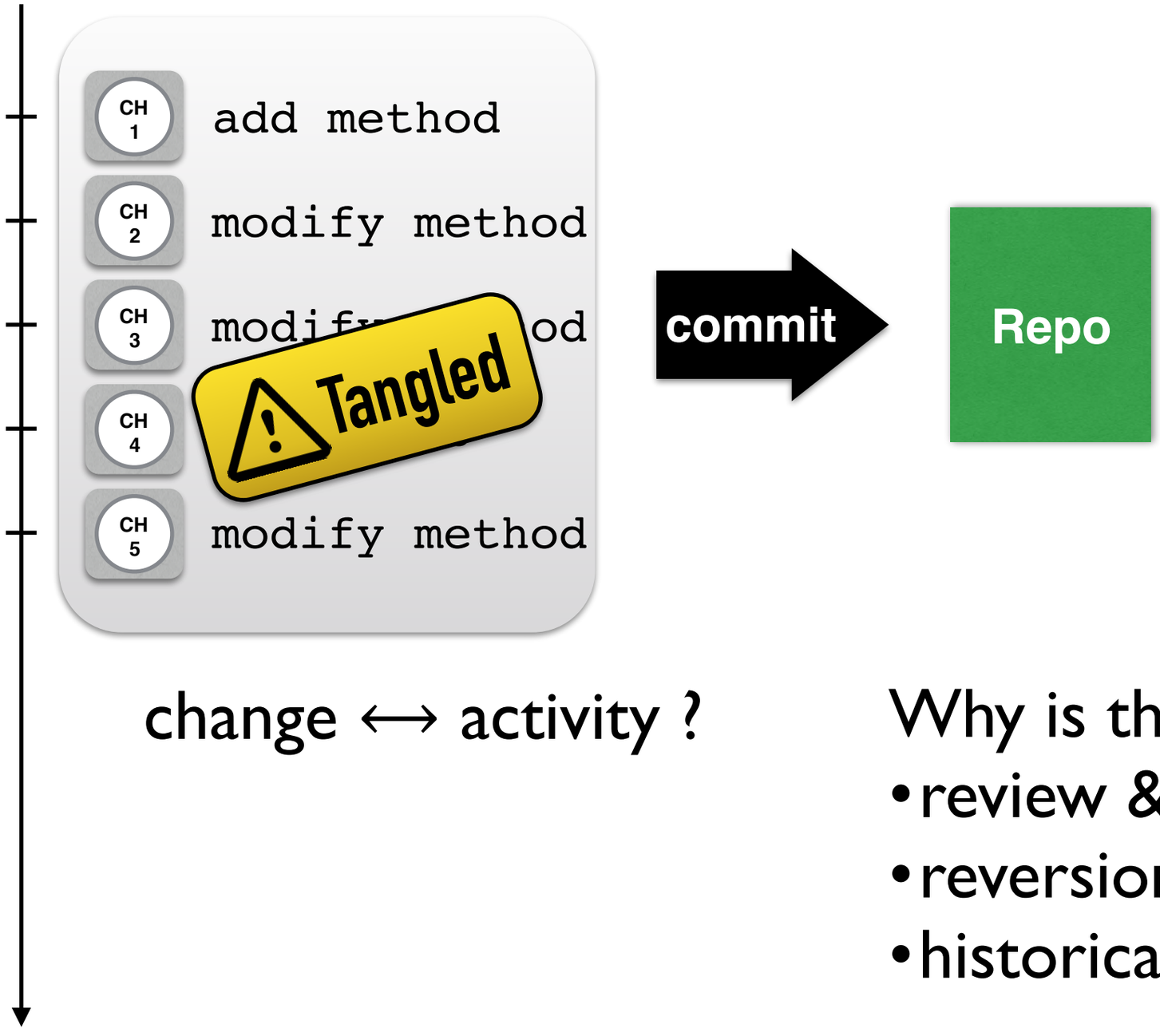
CH 4 add class

CH 5 modify method

commit

Repo

M. Dias



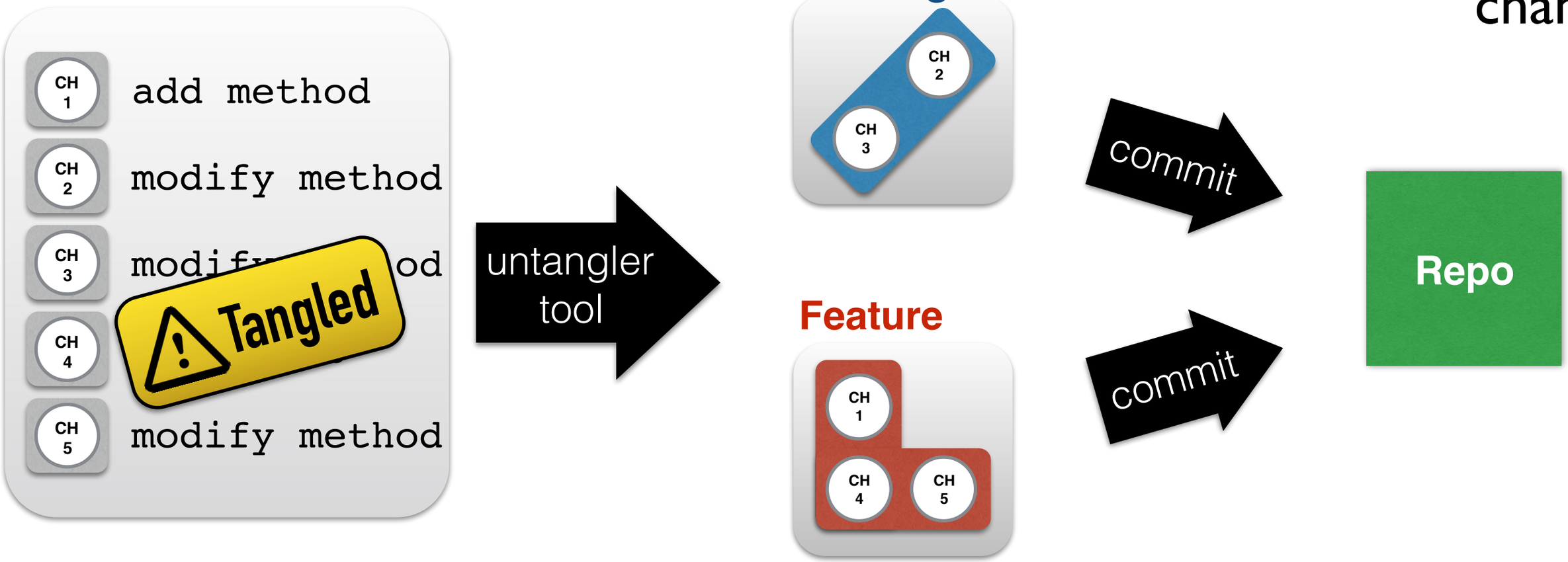
change ⟷ activity ?

Why is that bad?

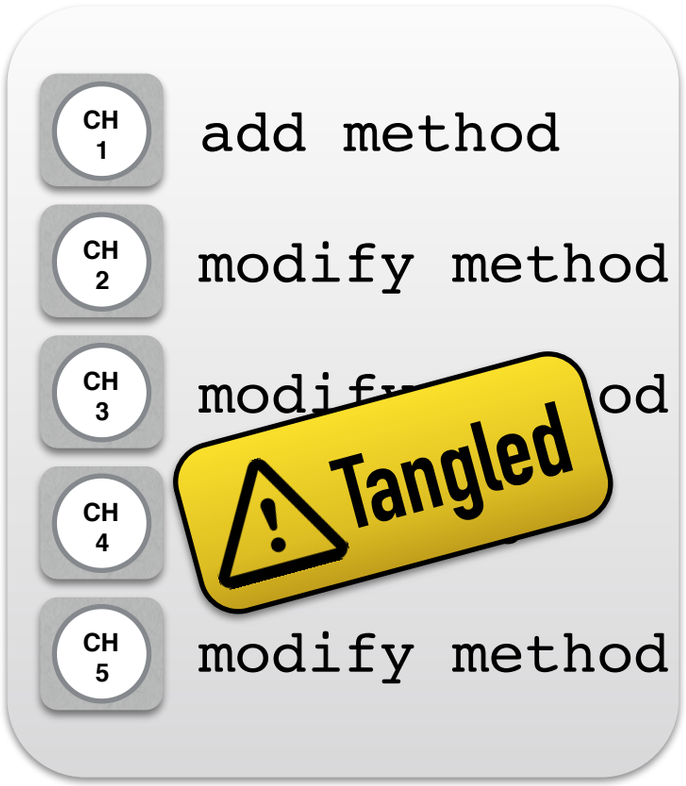
- review & integration
- reversion
- historical analysis

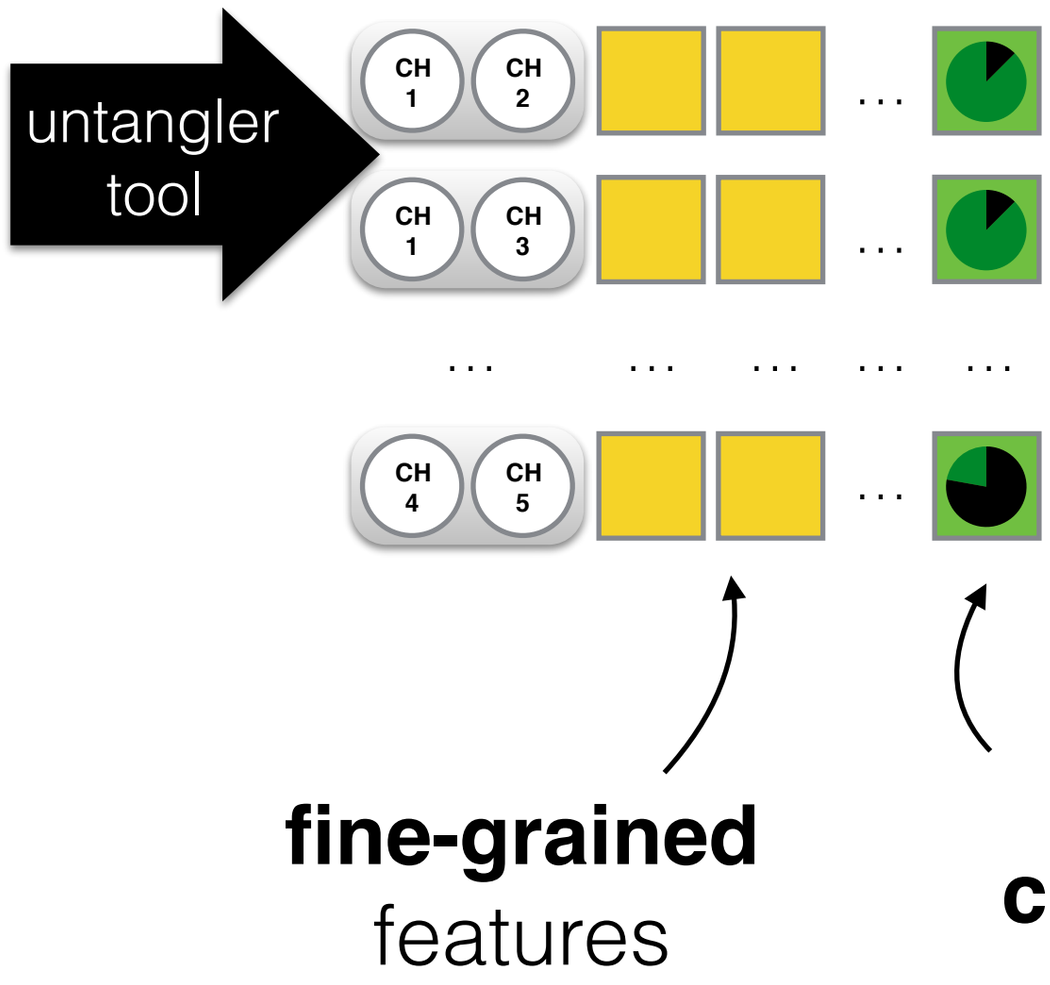
best practice: create untangled commits

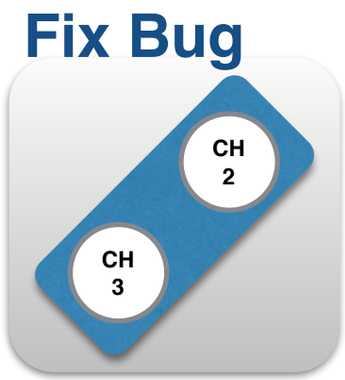
We want a tool to untangle fine-grained changes

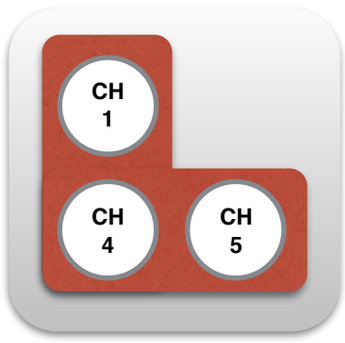


We want a tool to untangle fine-grained changes





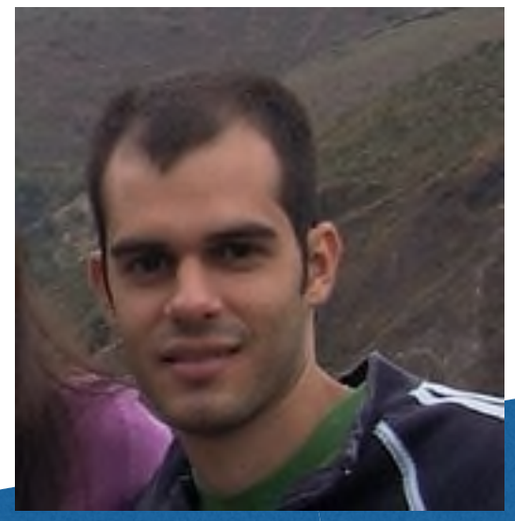




[SANER'15]

Supporting evolution

- Due to their cost, are system-specific rules worthwhile?
- Are rules good bug identification?



A. Hora

Automatically inferring migration rules

- Using developer activity
- Extract relevant API changes and deprecation rules

Reflective features

Pharo simple programmer expects…

- Incremental recompilation and changes
- Dynamic class shape changing
- Instance updates / migration
- Debugger support
- On demand stack reification

On the fly compilation

When a method is not found,

ask for the creation of a method on the fly

the system compiles on the spot a special method, then reexecutes the method

it raises a shouldBeImplemented exception

then you can edit the method in the debugger

then proceed and the program continues to run

Execution Stack as an Object

- Debugger support!
- Advanced debugging
- Continuation

thisContext

returns an object that represents the method activation

can walk and modify the stack

Halt

TextMorphForEditView(TextMorph)>>handleEdit:
PluggableTextMorph>>handleEdit:
PluggableTextMorph>>doIt
UndefinedObject(Object)>>perform:orSendTo:
[] in ToggleMenuItemMorph(MenuItemMorph)>>invokeWithEvent:
BlockClosure>>ensure:
CursorWithMask(Cursor)>>showWhile:
ToggleMenuItemMorph(MenuItemMorph)>>invokeWithEvent:
ToggleMenuItemMorph(MenuItemMorph)>>mouseUp:

Proceed | Restart | Into | Over | Through | Full Stack | Run to Here | Wher

```
invokeWithEvent: evt
    "Perform the action associated with the given menu item."

    | selArgCount w |
    self isEnabled ifFalse: [↑ self].
    target class == HandMorph ifTrue: [(self notObsolete) ifFalse: [↑
self]].
    owner ifNotNil:[self isStayUpItem ifFalse:[
        self flag: #workAround. "The tile system invokes menus
straightforwardly so the menu might not be in the world."
        (w := self world) ifNotNil:[
            owner deleteIfPopUp: evt.
            "Repair damage before invoking the action for better
feedback"
            w displayWorldSafely]].
```

self
all inst var
bounds
owner
submorphs

thisContext
stack top
all temp vars
evt
w
selArgCount

[326@121 mouseUp
42529 nil]

Powerful breakpoints?

"Stop method bar ***only if it is*** invoked from method testBar"

```
bar()

    ...

    this.haltIf(#testBar)

    ...
```

```
foo()
    this.bar()
```

Executing foo does ***not*** stop

while executing testBar ***should stop***

```
haltIf: aSelector
        | cntxt |
        cntxt := thisContext.
        [ cntxt sender isNil ]
                whileFalse: [
                        cntxt := cntxt sender.
                        (cntxt selector = aSelector)
                            ifTrue: [ Halt signal ] ]
```

pointer swapping

anObject become: anotherObject

All the pointers pointing to anObject points now to anotherObject and the inverse atomically

```
| pt1 pt2 pt3 |
pt1 := 0@0.
pt2 := pt1.
pt3 := 100@100.
pt1 become: pt3.
self assert: pt2 = (100@100).
self assert: pt3 = (0@0).
self assert: pt1 = (100@100).
```

Some challenges

Bootstrap reflective kernels

Update core libs at runtime

Virtualization (debugging our brain)

Isolate parts

Layered reflective representations

Multiple language runtimes side by side

JIT optimizations ready at startup

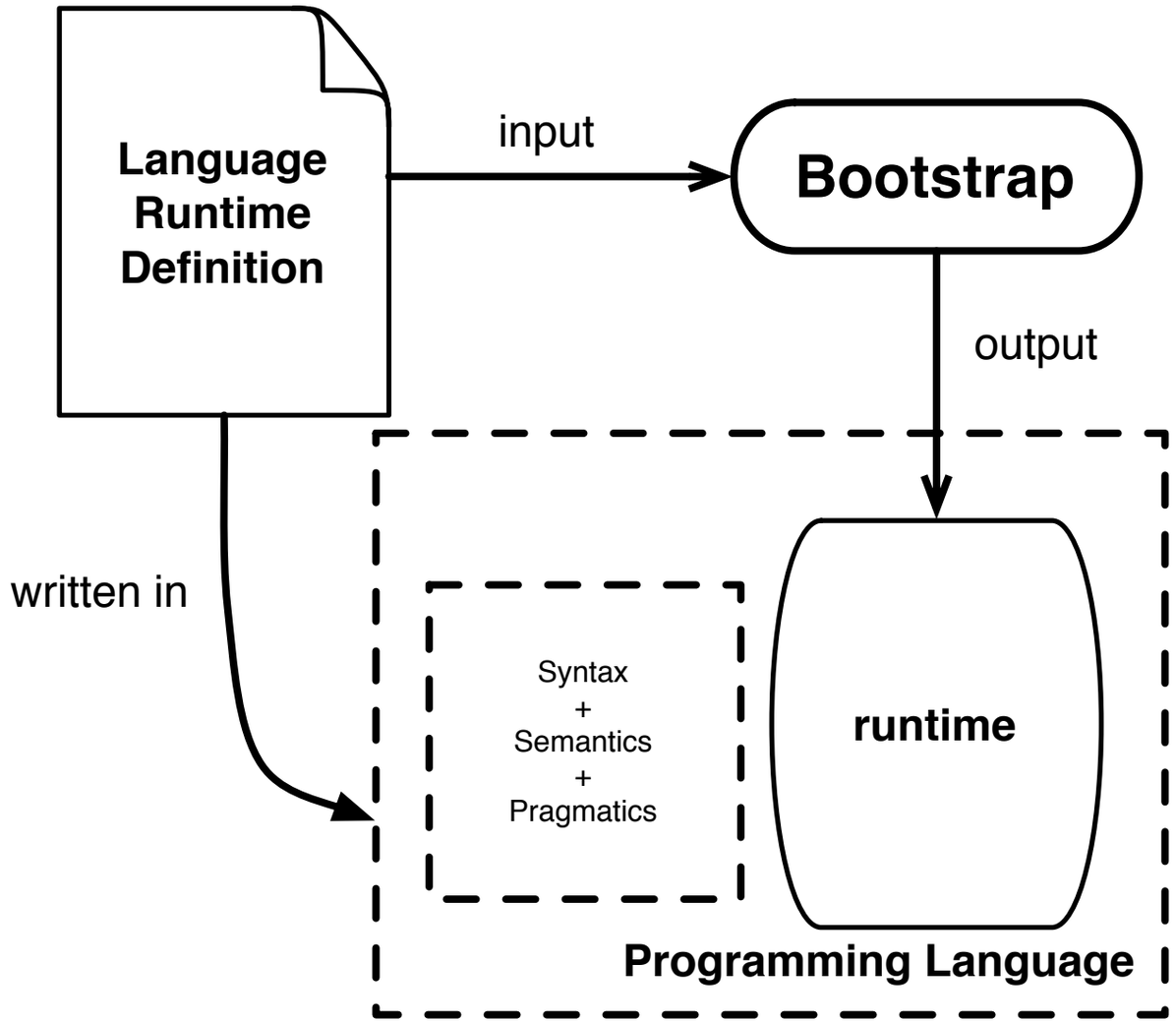
- How can we bootstrap kernels?
- How can we tailor and build specialized runtimes?
- How can we update core libs at runtime?
- Presentation @ Onward!

G. Polito

Bootstrap Definition

Language runtime Bootstrap

A process whose *input* is the definition of a language runtime written in the same language, and whose output is this language's runtime.



Bootstrapping Process

1. Create well-known objects

```
nilObject := UndefinedObject basicNew.
trueObject := True basicNew.
falseObject := False basicNew.
```

2. Create basic language structures

```
globalTable := GlobalTable basicNew.
```

3. Creat

How do we execute this bootstrap?

```
ject
instanceVariableNames: ”.
```

4. Create methods

```
Object >> isNil
^ false

UndefinedObject >> isNil
 ^ true
```

5. Initialize

```
Float initialize.
Processor initialize.
```

Minimal specialised runtimes

By default any runtime is too fat :)

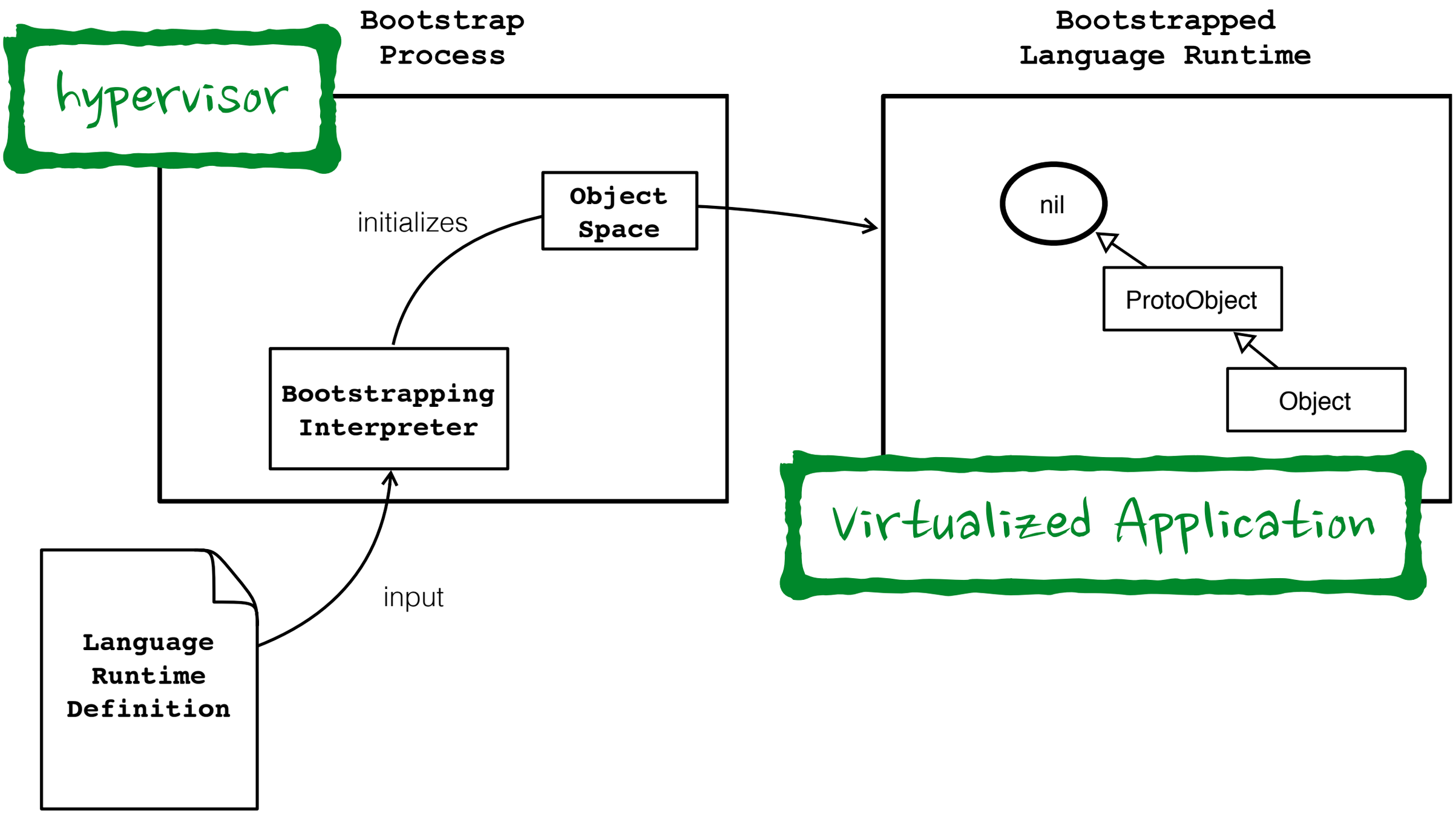
How can we tailor and build specialized runtimes?

How can we update the update while running the update?

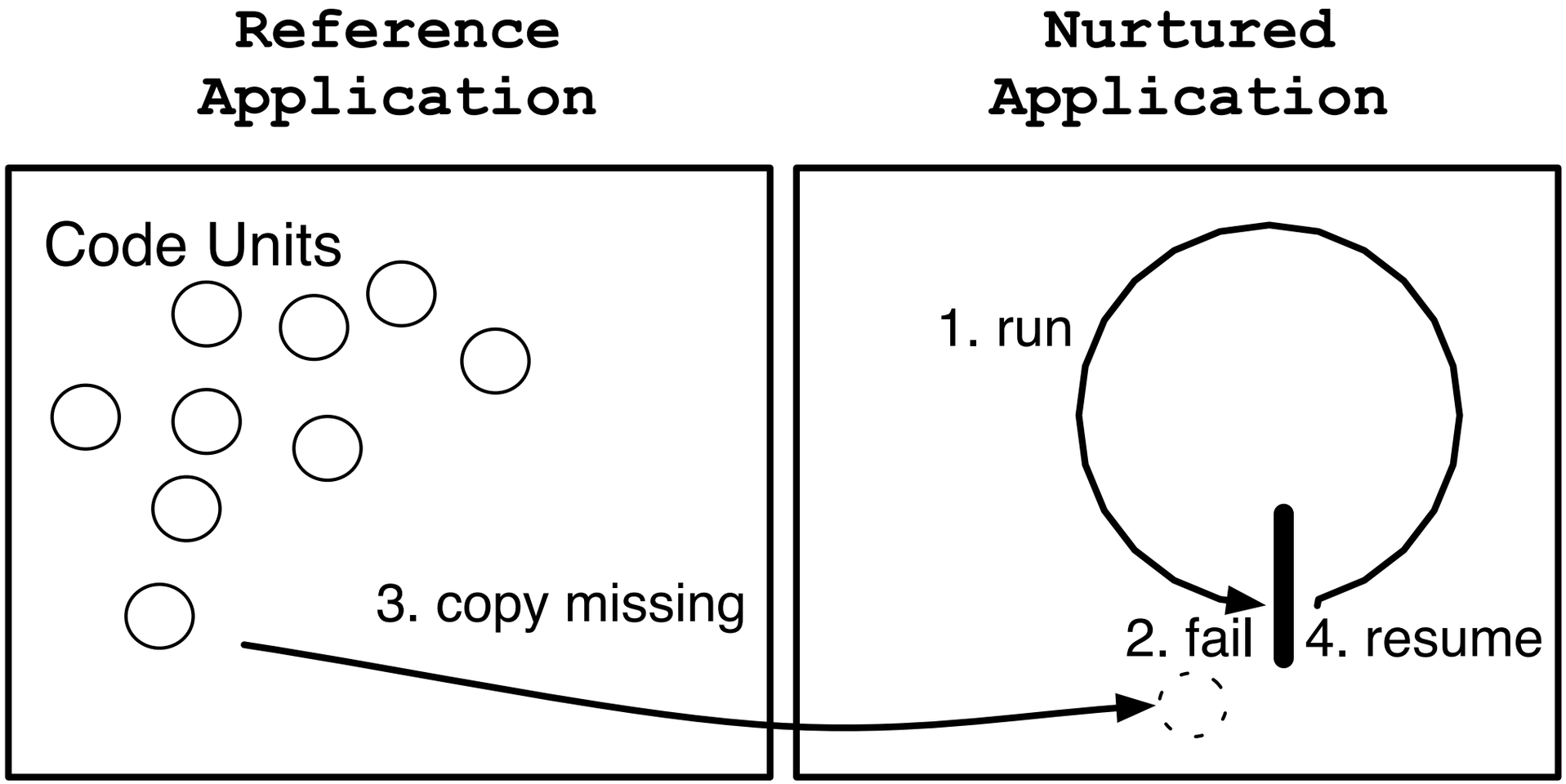
Examples:

- Dictionary hashes
- Delay refactoring
- Collection changes

Espell Virtualization Infrastructure



Espell again: Run-Fail-Grow



Results

Experiment	Size (KB)	%Saved
Addition	11	99.99%
Reflective App	32	99.83%
Factorial 100 + I/O	89	99.39%
Seaside Counter	573	96.73%

Experiments with empty seeds

Object Space

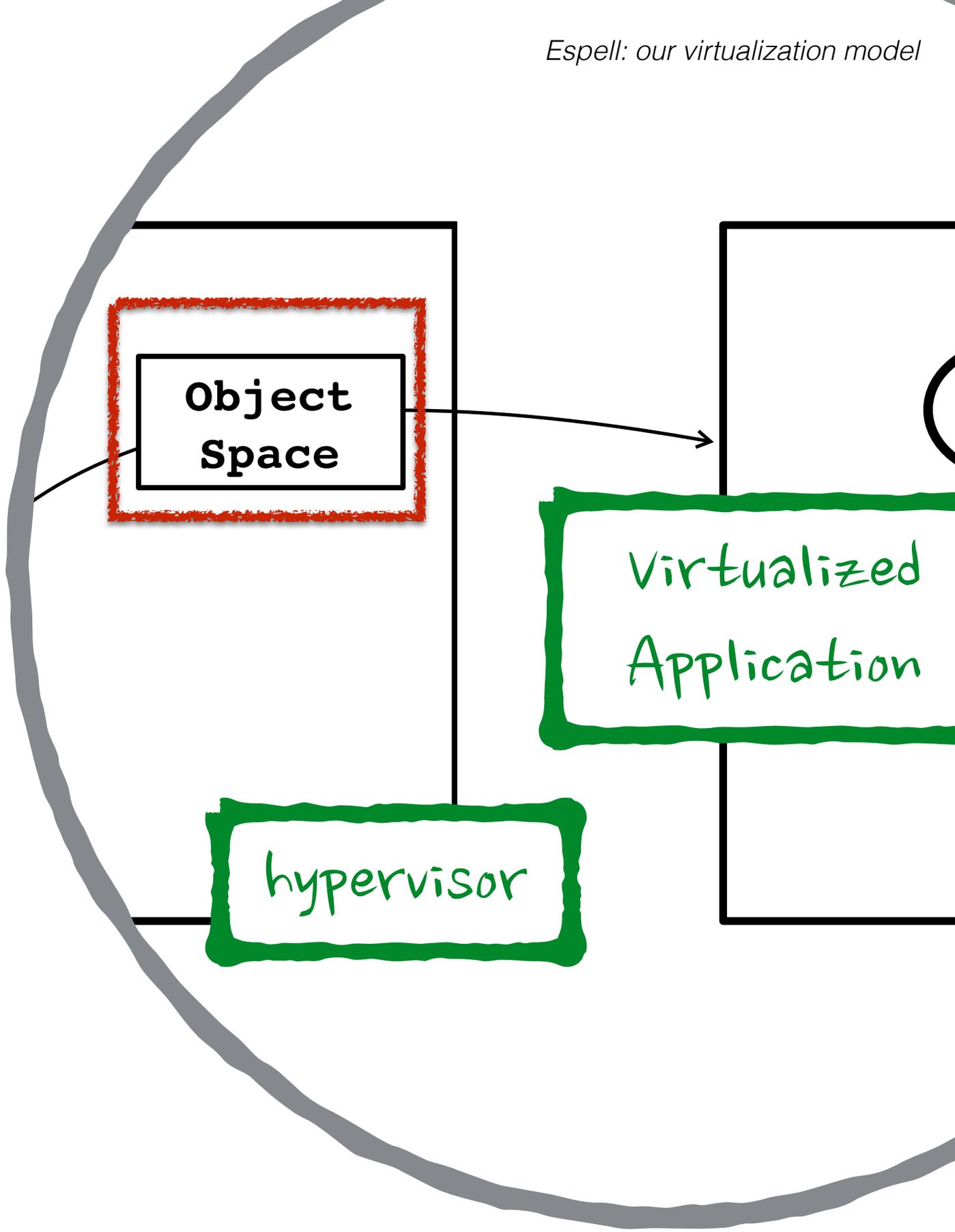
- First-class virtualized runtime
- It's a meta-object! => MOP

Language Manipulation
e.g., create class

Object Manipulation
e.g., set slot, get slot

Execution manipulation
e.g., create process, get stack frames

Espell: our virtualization model



Lowering JIT warm up time

Saving hot VM state

- Saving JIT compiler optimisations across start-ups
  - Saving compiled methods encoded with an extended bytecode set (no n-code)
- Time to reach peak performance avoided
- PhD of C. Béra co-supervized with Cadence Design Systems (US)

C.Béra

How to test dynamic deoptimisation?

Can we build regression tests for dynamic deoptimisation?

Using symbolic execution

- Byte-code
- Constrained to scope optimisation (do not execute not inlined calls)

Reflective architecture

How modular reflective architectures can be? Can we package language extensions?

How to reconcile reflection with object encapsulation (and object capabilities)?

Stratification

- Conflation of high-level concerns (language) and low-level concerns (the VM):
- Language extensions are not modular
- Examples:
  - Reflection reveals implementation details of language extension
  - Generated code should not be visible via an introspective API

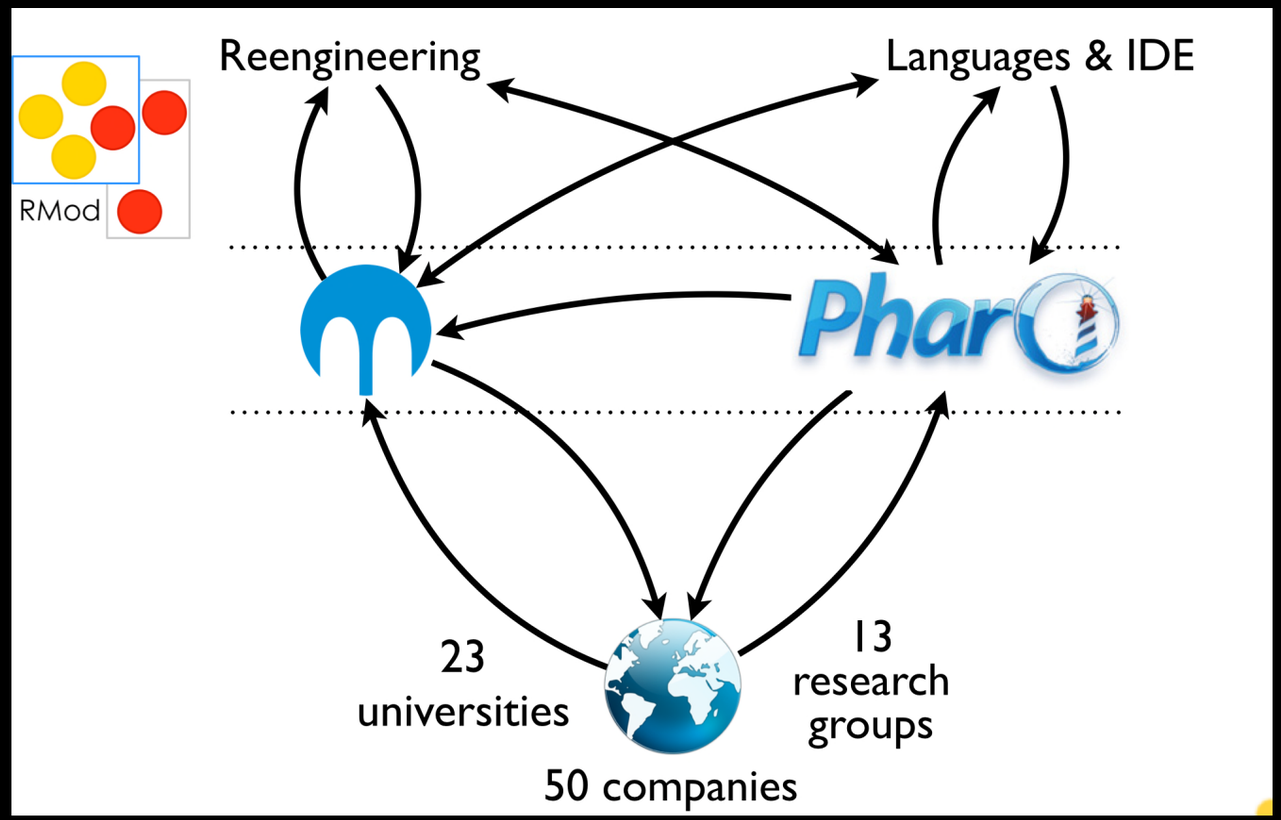
Encapsulation *vs.* Reflection

Reflection breaks object encapsulation. This makes reflection and object capabilities incompatible.

What kind of access control can prevent abuses while maximising the power of reflection?

Ownership-based access control

Happily
building platforms & communities




THANKS!