Pharo for Scientists

http://stephane.ducasse.free.fr
http://www.pharo.org

JPMorgan Chase & Co.

ENSTA Bretagne

Lam Research

Inria
INVENTEURS DU MONDE NUMÉRIQUE

feenk

SCHMIDT
Ingenieurbüro für Bauwesen

Lifeware

ZWEIDENKER

HR works

THALES

GemTalk Systems

Software Soops Group

projector software

BetaNine
software engineering

netstyle.ch

inspired!

SORABITO

knowroaming

TA MÈRE scrl
Badass Mobile development

FINWORKS
INFORMATION TECHNOLOGY SOLUTIONS

Systems that make sense
Sensus

LABWARE
Results Count

OBJECT PROFILE

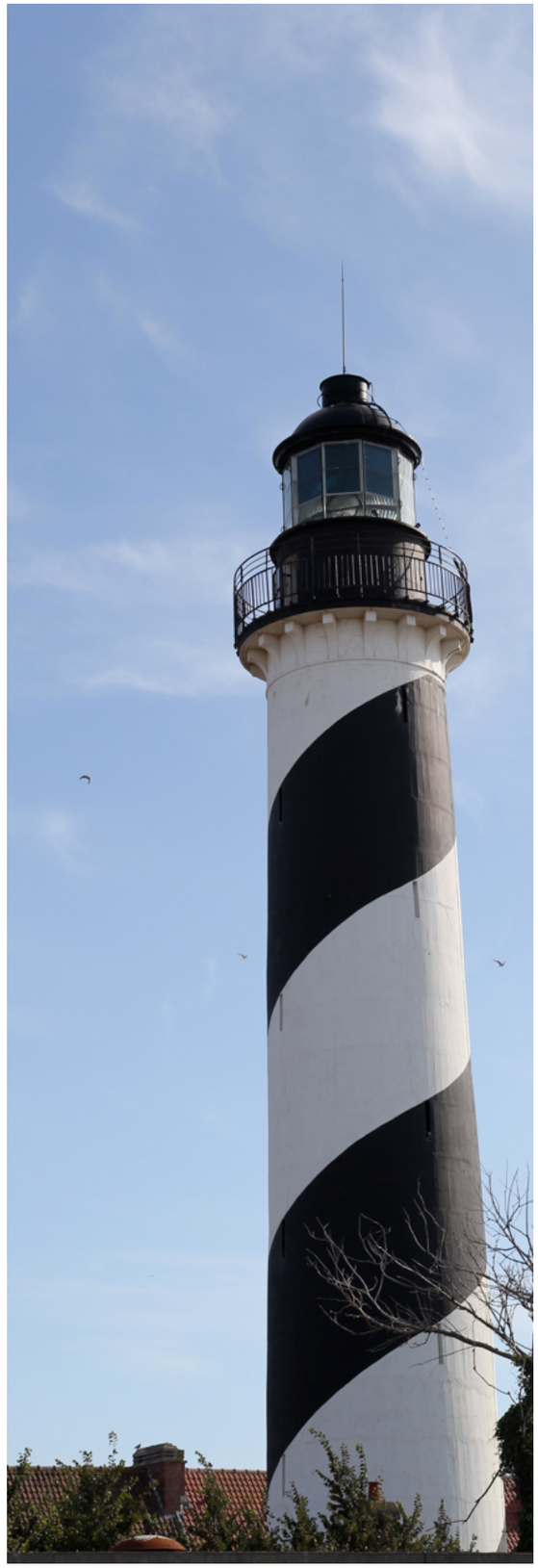
Palantir

SpesenFuchs

FAST
Fundación Argentina de Smalltalk

Yesplan
Let's make it happen

esug
EUROPEAN SMALLTALK USER GROUP



Pharo?

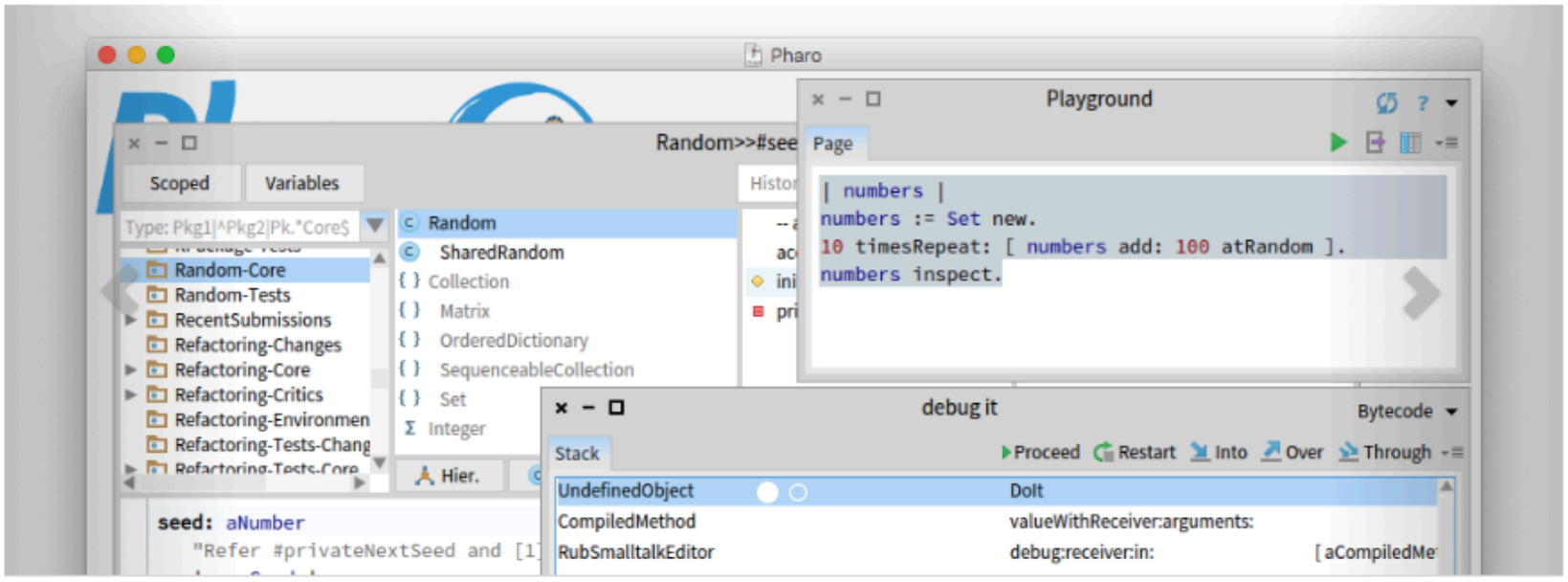
Pharo!

- System: Pure object language + full IDE
- Powerful, elegant and fun to program
- Great community / industrial consortium
- Living system under your fingers
- Works on Mac OSX, Linux, iOS, *Android*, Windows, Pi - 100% MIT

Pharo

The immersive programming experience

Pharo is a pure object-oriented programming language *and* a powerful environment, focused on simplicity and immediate feedback (think IDE and OS rolled into one).



Pharo 90
~740 packages
- 9 000 classes
- 120 000 methods

250 forks sur Github
up to 100 contributors
30 regulars
- 8 sub projets
  - graphics
  - vcs
  - tools

Consortium
~ 28 companies
~ 25 academic

Discover

Learn more about Pharo's key features and elegant design.

Download

Download latest version (8.0)!
Read more about here

Learn

Access the Pharo Mooc!
3000 people registered and follow the Pharo Mooc. You can find it here.
Watch the teaser!

Subscribe to the Pharo Newsletter

email address | Subscribe

Follow us on Twitter: @pharoproject

Elegant!

- Full syntax on a postcard
- Simple and powerful objet model

```
exampleWithNumber: x

	<syntaxOn: #postcard>
	"A ""complete"" Pharo syntax"
	| y |
	true & false not & (nil isNil)
	 ifFalse: [ self perform: #add: with: x ].

	y := thisContext stack size + super size.

	byteArray := #[2 2r100 8r20 16rFF].

	{ -42 . #($a #a #'I''m' 'a' 1.0 1.23e2 3.14s2 1) }
	 do: [ :each |
		| var |
		var := Transcript
		 show: each class name;
		 show: each printString ].

	^ x < y
```

method name
parameter
pragma
comment
local variable
binary message
unary message
boolean literals
nil literal
block
keyword message
assignment
pseudo variables
instance variable
integer literals
byte array
array generated at runtime
literal array
symbols
floating point
character
string
scaled decimal
local block variable
block parameter
global variable
cascade
keyword message
return instruction

other method definition examples:
unary
+ binaryMessageArgument
keyword: arg
keyword: arg1 withTwo: arg2

Full Syntax!

https://www.pharo.org

Mainly objects, messages and lexical closures

Temp declaration, assignment, separator and return

```
| a |

a := 40 + 2.

^ a
```

Lexical closures: Blocks

Unary messages

Date today, 9 squared

“Operators” Binary

4 + 3

‘Black ’, ‘Chocolate ’, ‘is good’

Keywords messages

2 between: 0 and: 10

2.betweenAnd(0, 10)

Lexical closures: Blocks

- closure definition

```
[:x | x + 5]
```

- closure execution

```
[:x | x + 5] value: 37

>>> 42
```

Control flow ops are messages

```
Weather today isRaining
    ifTrue: [ self takeUmbrella]
    ifFalse: [ self takeSunglasses]
```

Yes we send messages to Boolean objects

Iterators

#(1 -2 -3 4 -5) collect: [:each | each abs]

>>> #(1 2 3 4 5)

Yes we send messages to collection to do loops

Class / Method definition

```
Object << #Point
    slots: { #x . #y };
    package: ‘Kernel'


<= aPoint
    "Answer whether the receiver is neither below nor to the right of aPoint."
    ^ x <= aPoint x and: [y <= aPoint y]
```

Full Model!

- Dynamically typed
- Everything is an instance of a class
- All methods are public virtual
- All attributes are protected
- Single inheritance
- Stateful Traits
- Closures everywhere

For the advanced

- Classes are objects
- Instance variables are first class
- MOP for read/write
- Stack can be reified on demand
- AST can be fully annotated and transformed on the fly during execution

Focus on intrinsic **complexity**

Eliminate accidental **complexity**

Fully Written in Itself

Everything is written in Pharo
One simple syntax to rule them all

Learn
Adapt
Modify

Pharo is highly immersive

Most languages do not expose much of their internals

Looking inside

- Introspection is a first step
- Pharo is fully inspectable and reflective

jonrawlinson.c

Immersed and interacting

- You interact with objects
- You modify objects life

Workspace

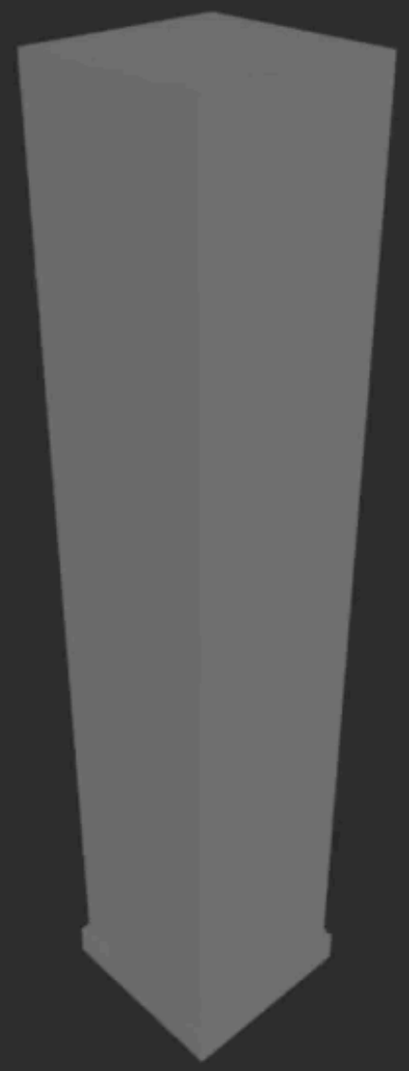
```
| elements lay |


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


lay := R3WallLayout new.
lay on: elements.

UberPresenter present: elements
```

Uber Presenter



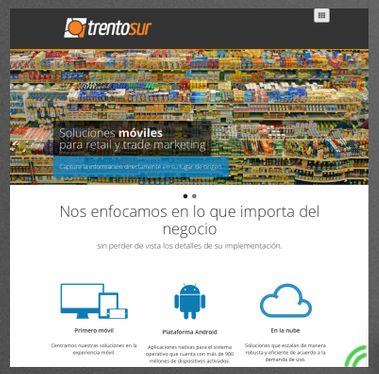
We can do the same with web app, sockets, networks, sensors, living programing....

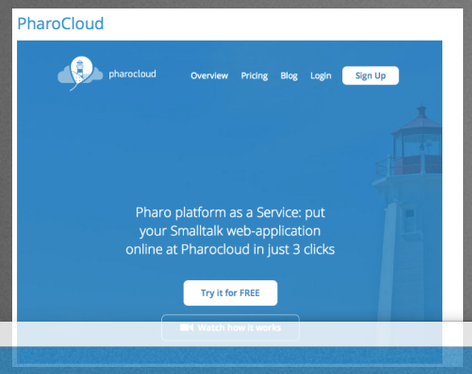
Pharo's goal...

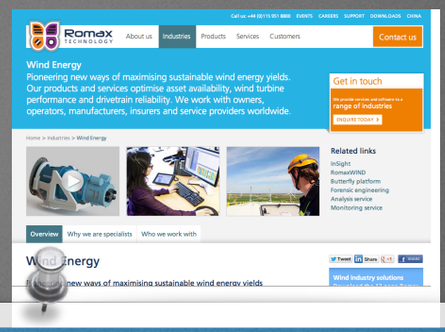
A powerful engine
to invent (y)our future

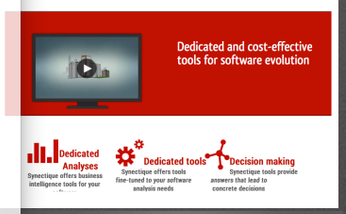
An ecosystem where innovation/business bloom

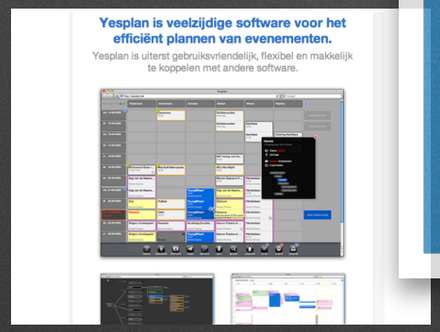
Some Success Stories

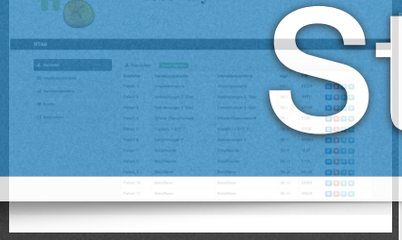


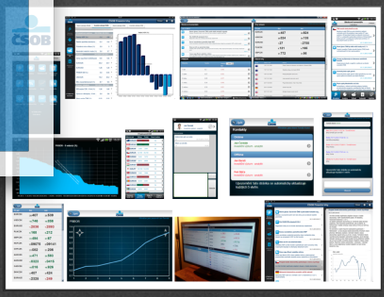


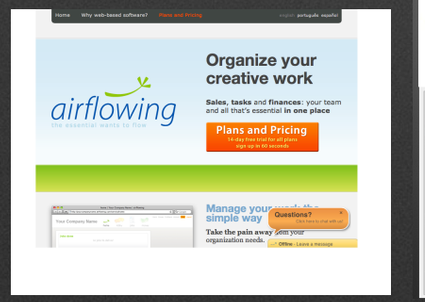


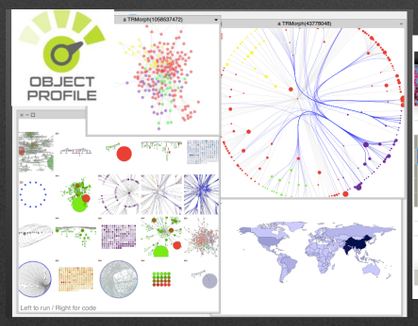


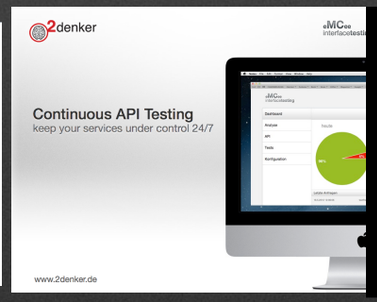


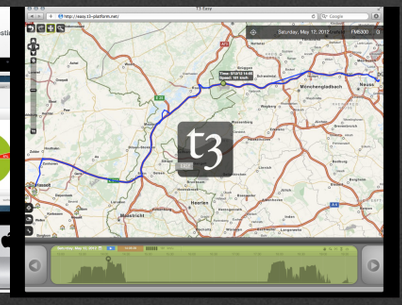


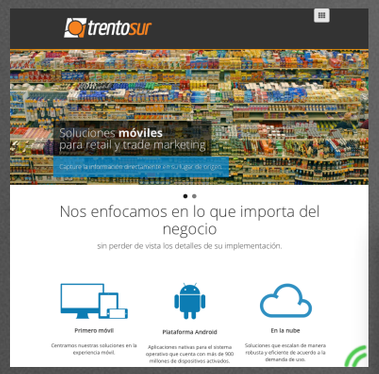


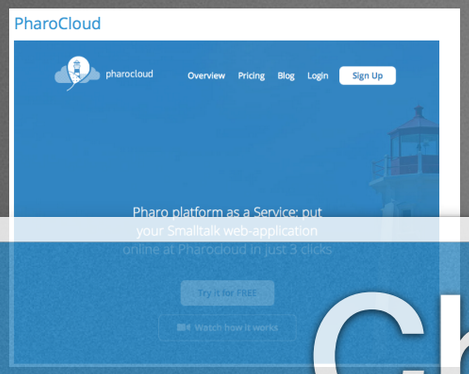


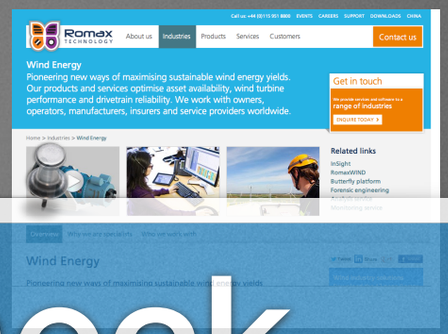


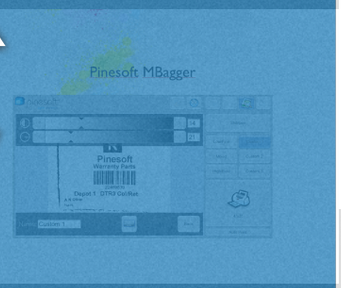


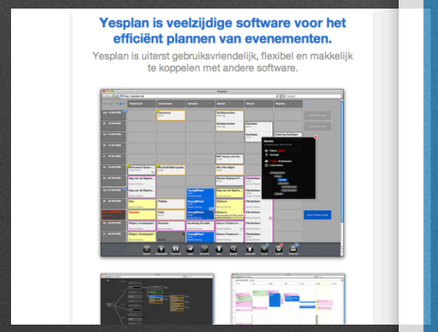


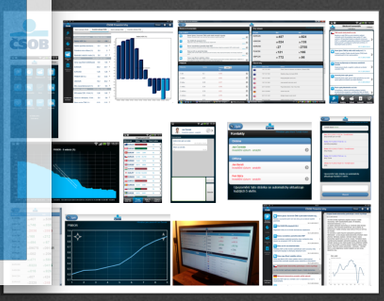


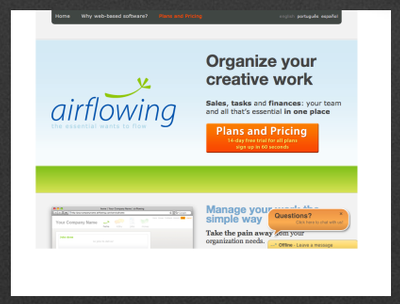


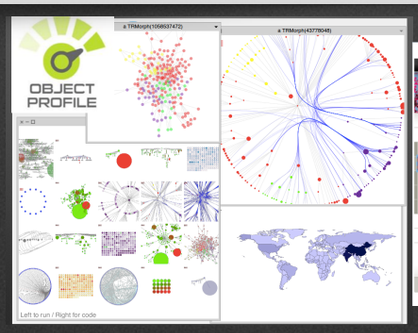


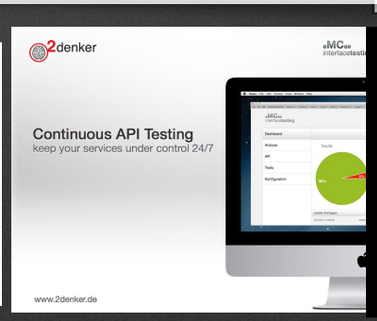


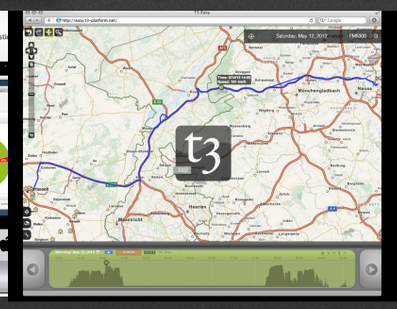








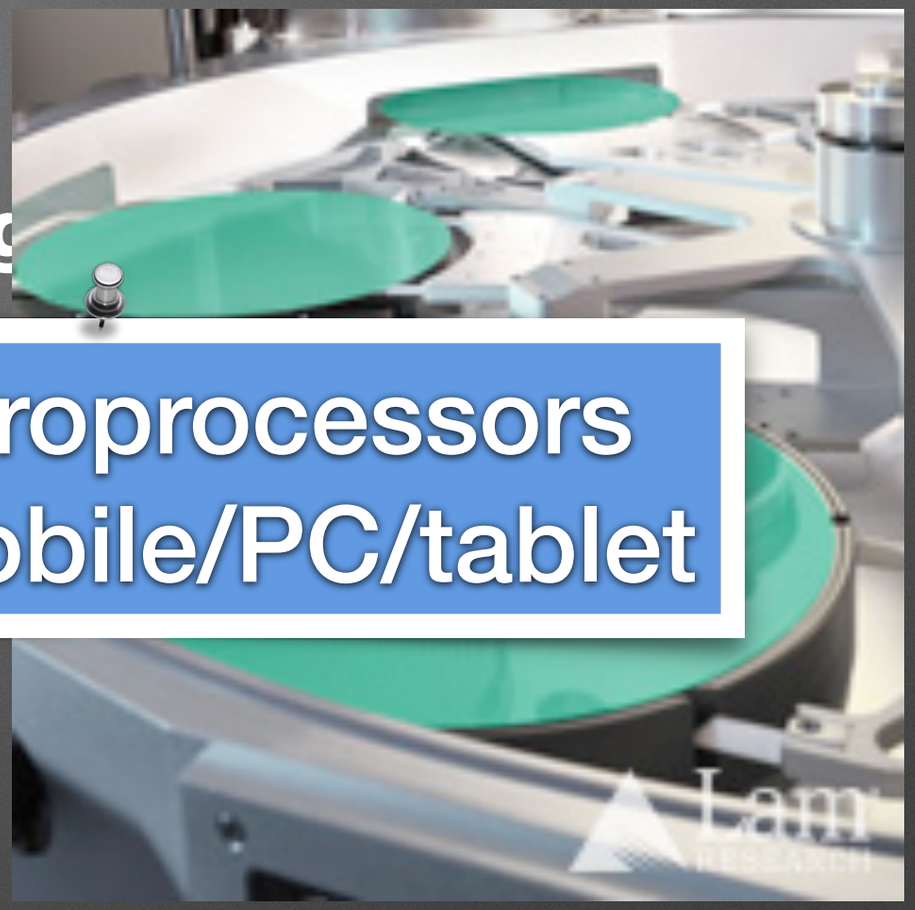




Check http://pharo.org/success

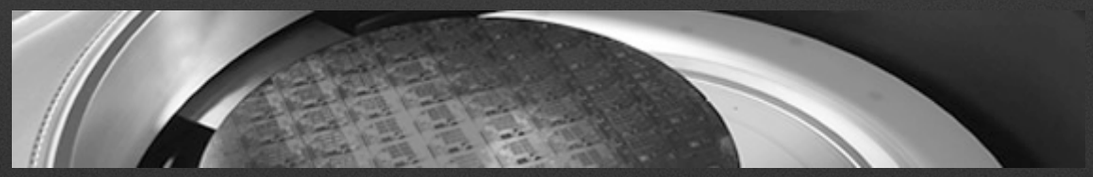
LAMRC

Back-End Wafer-Level Packaging



50% microprocessors
of your mobile/PC/tablet

Your mobile hardware as
a *high-chance* to have
been produced by a robot
controlled via a smalltalk

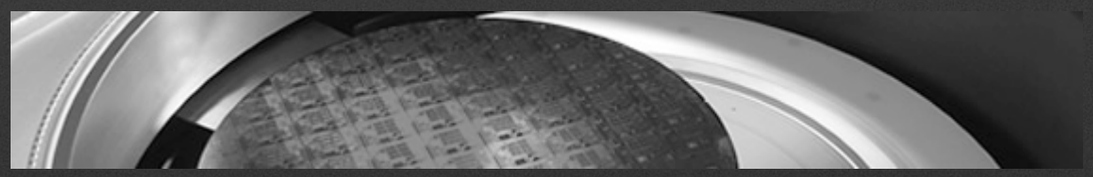


LAMRC

Back-End Wafer-Level Packaging

Started to experiment with Pharo

Your mobile hardware as a high-chance to have been produced by a robot controlled via a smalltalk



LifeWare

Lifeware

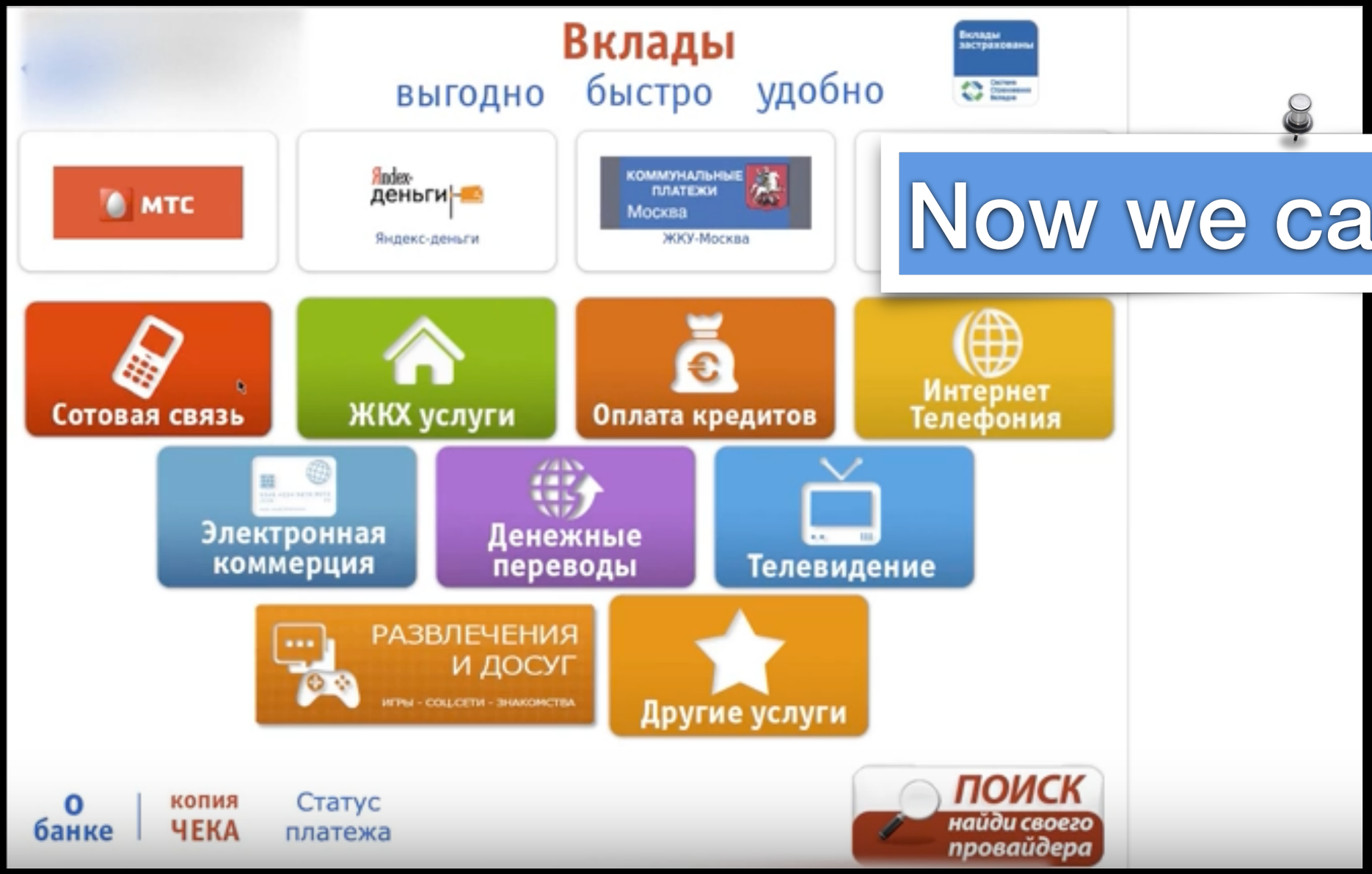
800 thousand	policies currently managed by our solution
25 € billion	assets under management
15	satisfied customers
90 thousand	page per views per day
130 thousand	authorized users

Migrating to Pharo

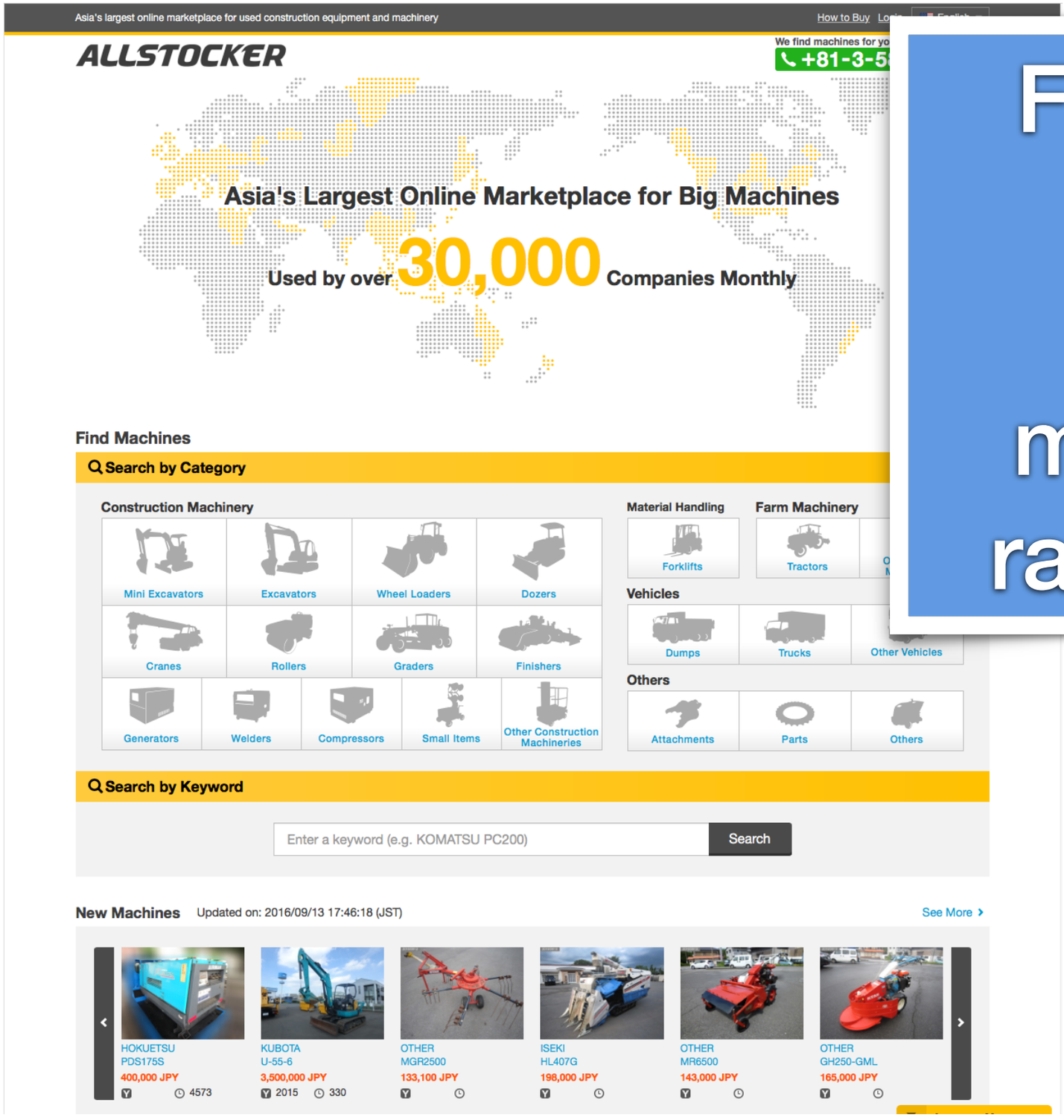
30 Millions lines of code…

Kent Beck is one of the founders

Pharo in ATM like Device in Moscow



Now we can say it



Full Stack in Pharo Online marketplace raised 4.7 M$

Network Monitoring for small ISPs

pharocloud

Overview Pricing Help&Docs Log In Sign Up

Pharo platform as a Service: put your Smalltalk web-application online at Pharocloud in just a couple clicks

Try it for FREE

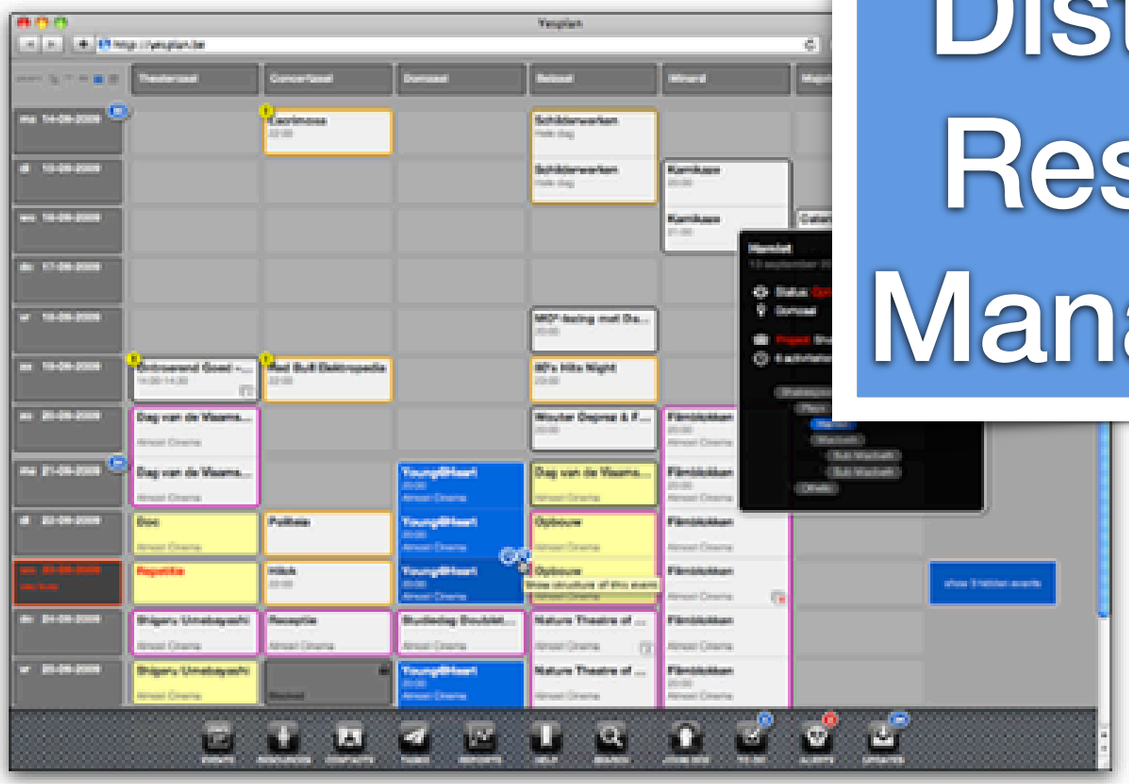
Watch how it works

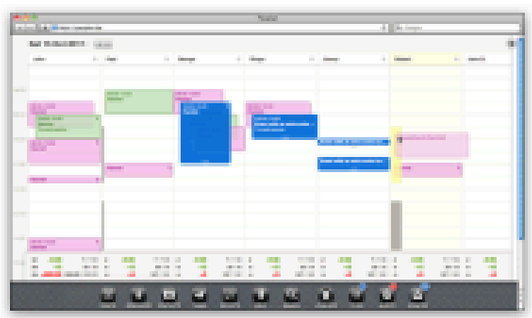
Deploy Pharo on the cloud in 3 min

CMSBOX
Bearbeiten
Hinzufügen
Clipboard
Aktionen
Diese Seite
sofort
Publizie
Neue Seite
Artikel
Formular
Tabelle
cmsbox
CMS with full history
SYSTEM
WEBSEITE
Bearbeiten
Metatags
Statistik
Archiv
BENUTZER
SUPPORT
Das Content Ma
100% Inline-Editor
Die cmsbox ist ein Content Management System der nächsten Generation. Die Benutzeroberfläche ist nahtlos in die Webseite integriert: Alle Texte werden direkt auf die Seite geschrieben!
Drag & Drop
Handlich: Boxen und Elemente frei auf der Seite positionieren.
mail@cmsbox.com
Copy / Paste
Elemente ausschneiden oder kopieren und irgendwo auf der Webseite wieder einfügen.
Text einfügen
cmsbox rocks
Eigene Bilder und Videos
direkt auf die Webseite la
CMSBOX GMBH · TERRASSENWEG 18 · CH-3012 BERN · +41 (0)31 356 42 52
Suchen…

Yesplan is veelzijdige software voor het efficiënt plannen van evenementen.

Yesplan is uiterst gebruiksvriendelijk, flexibel en makkelijk te koppelen met andere softw





Distributed Resources Management

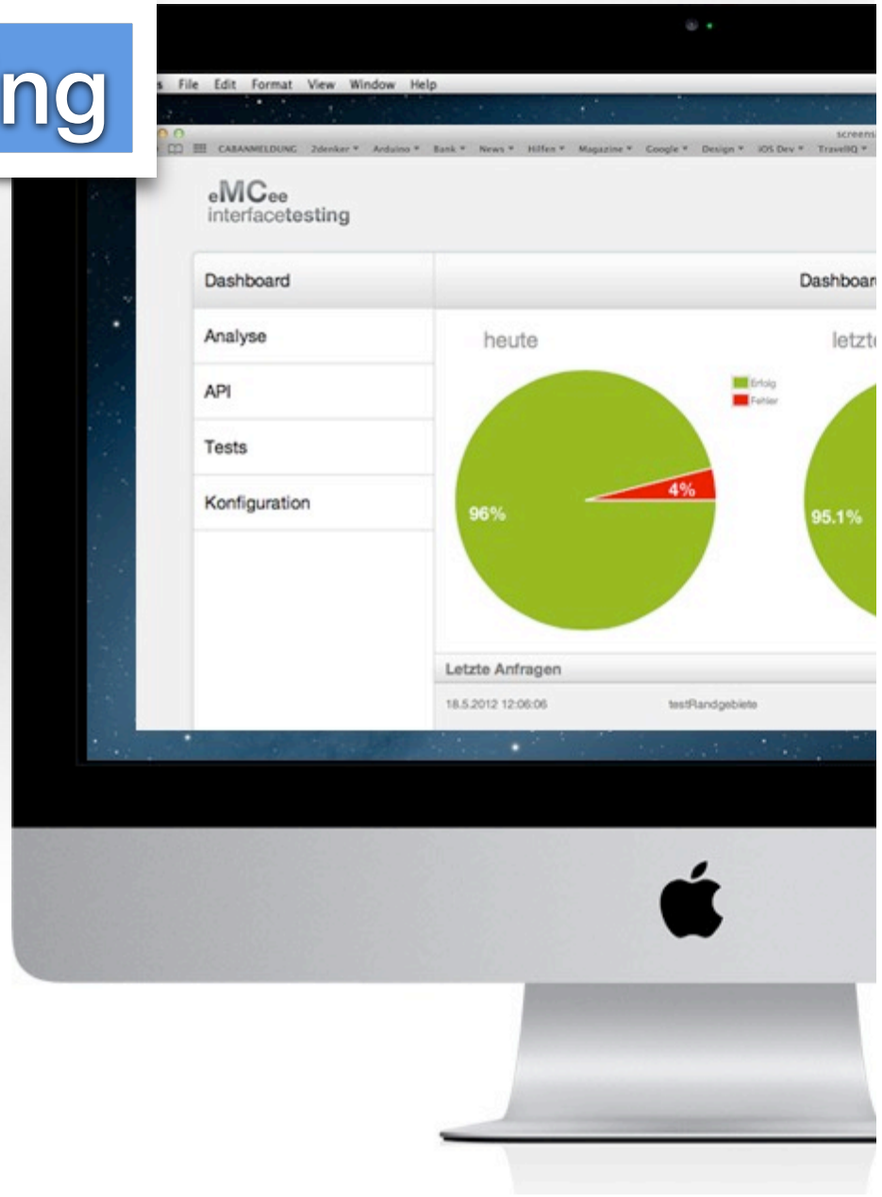
2denker

eMCee
interfacetesting

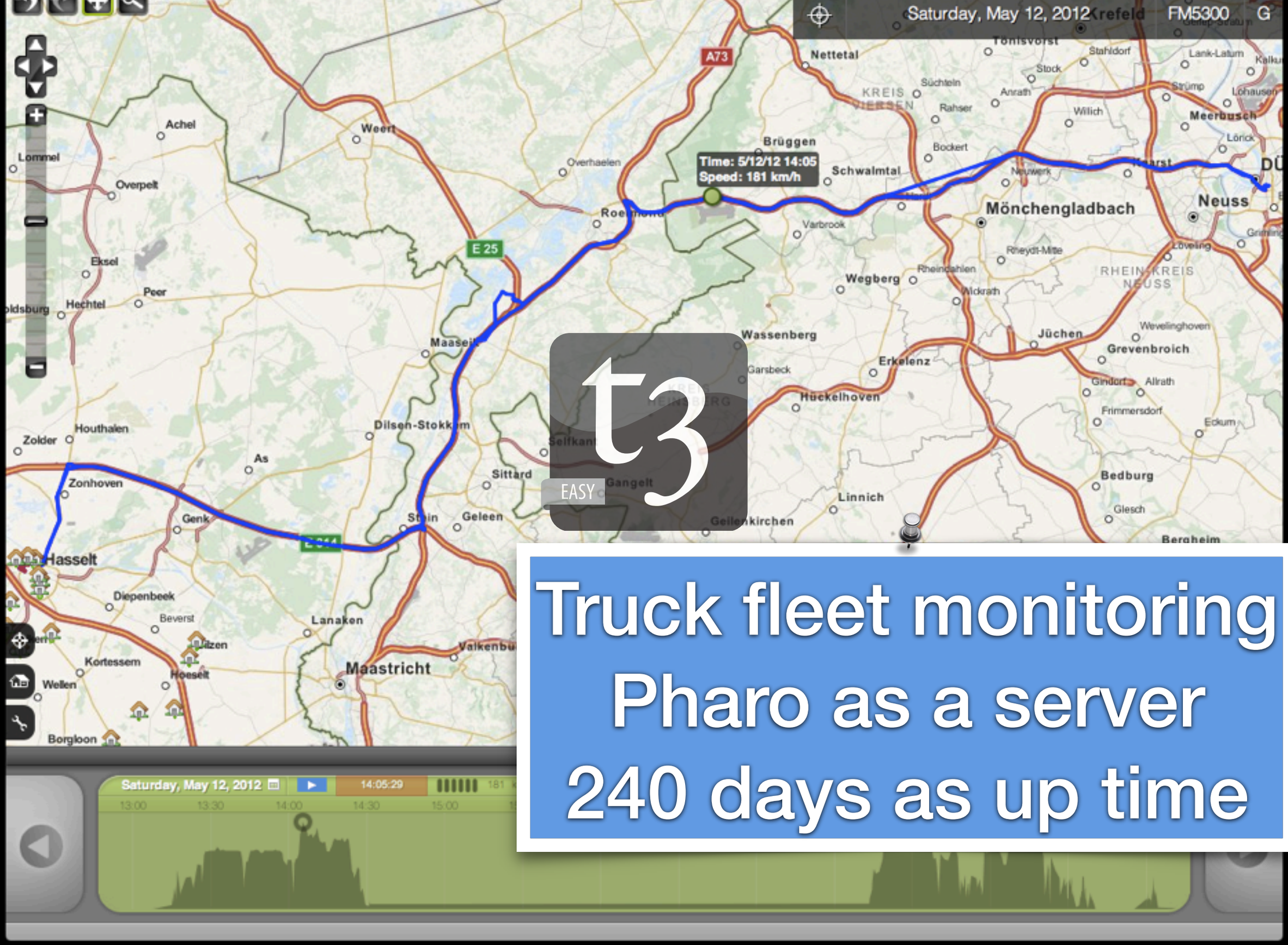
Server Testing

Continuous API Testing

keep your services under control 24/7



www.2denker.de



Truck fleet monitoring
Pharo as a server
240 days as up time

netstyle.ch

WEBDRUCK.CH
Web-To-Print Solution

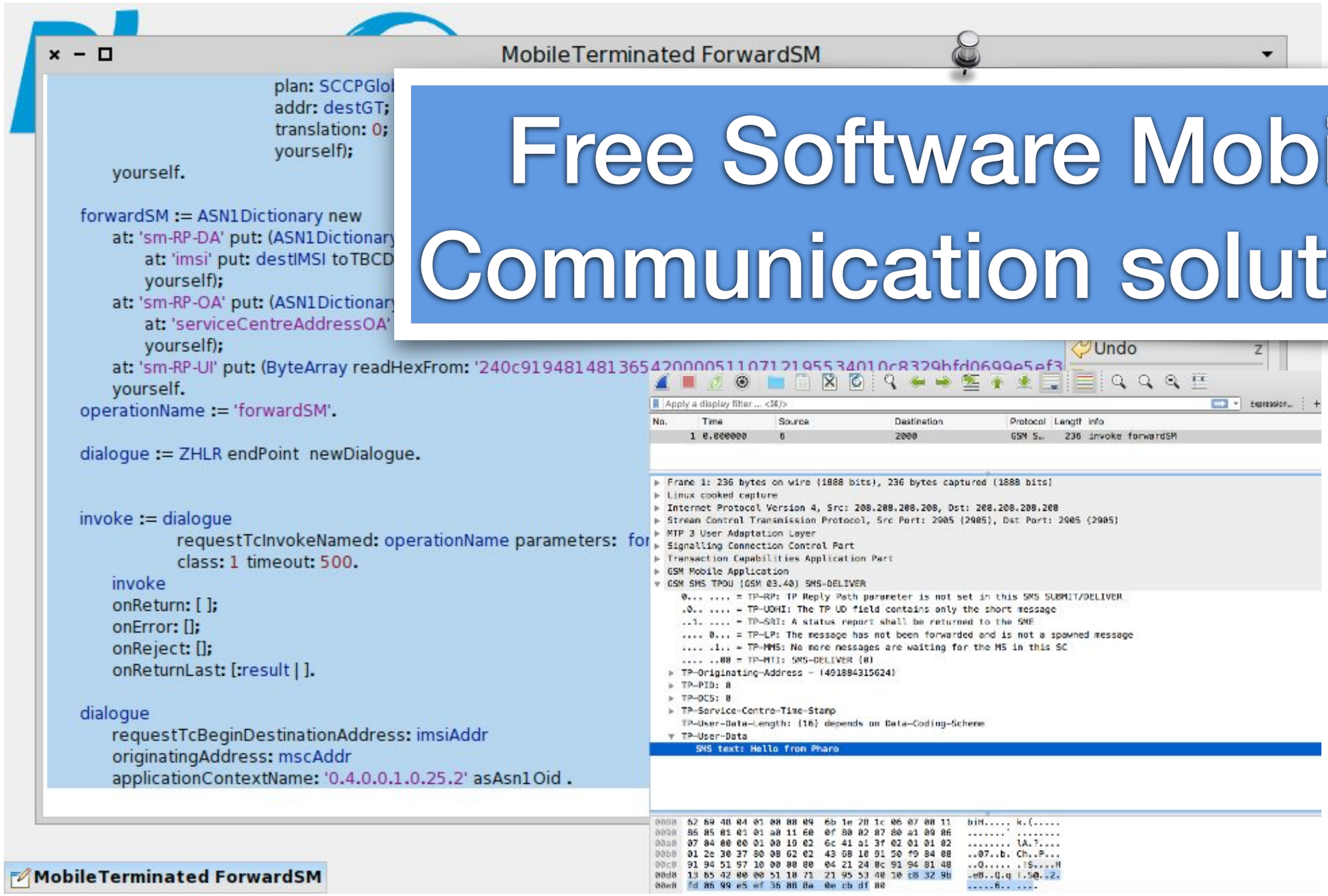
- Design and create individual printed matter
- eShop with credit card payment
- High quality PDF output with Printing Process integration
- Thousands of orders for seven Swiss printing companies

All reports are automatically generated

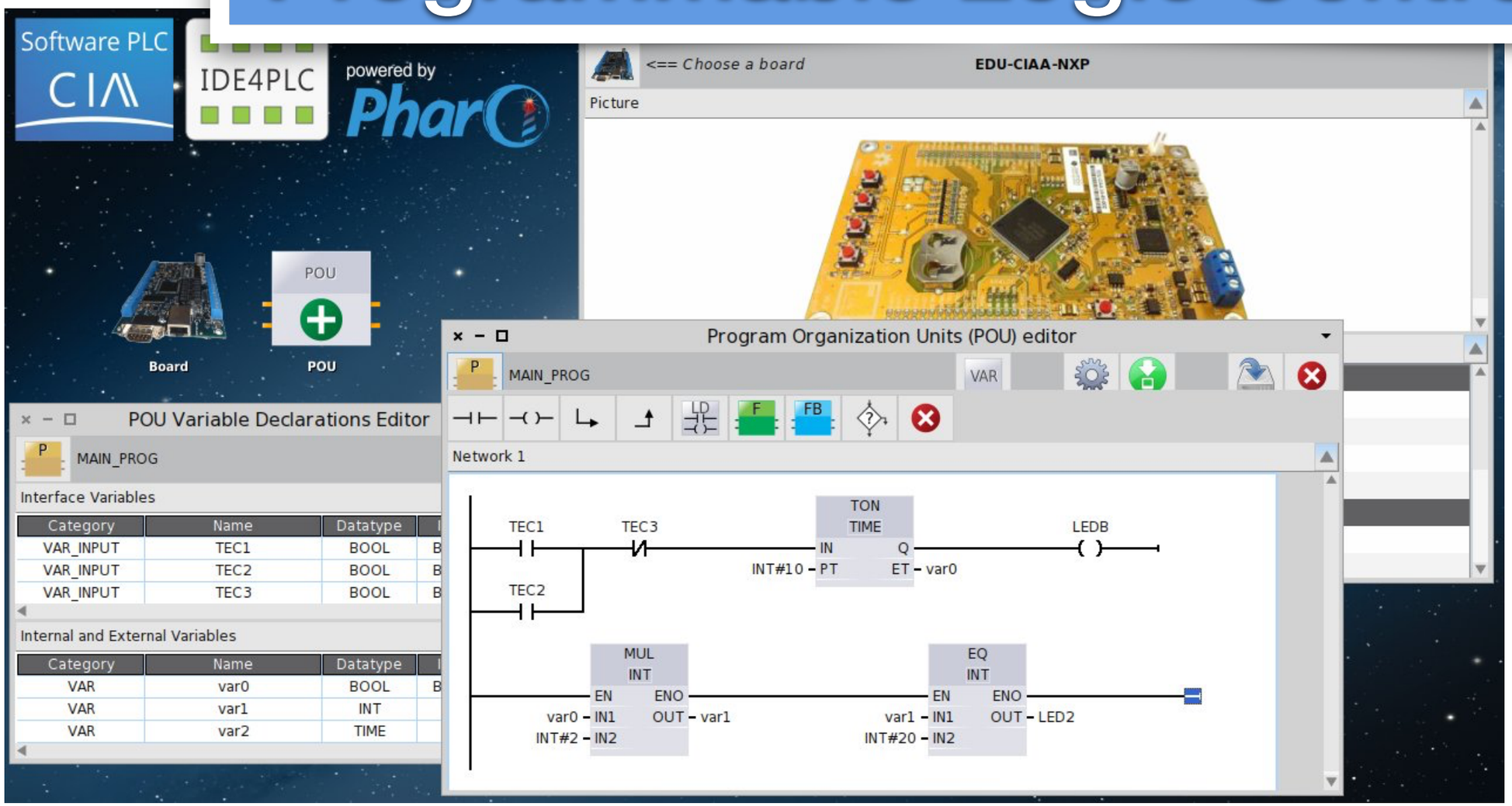
Quuve: Investment Portfolio

http://debrispublishing.com

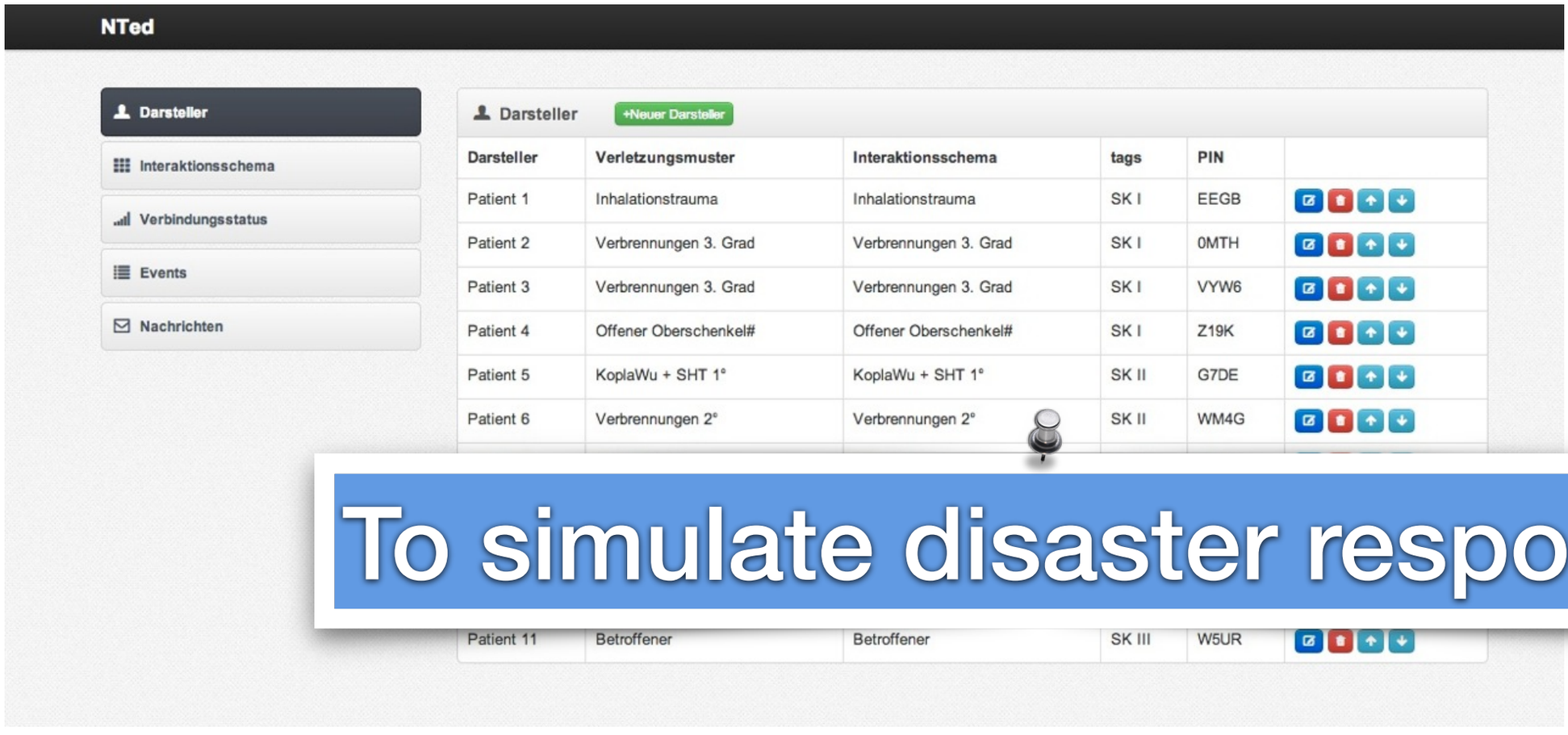
Free Software Mobile Communication solutions



Programming environment for Programmable Logic Controllers



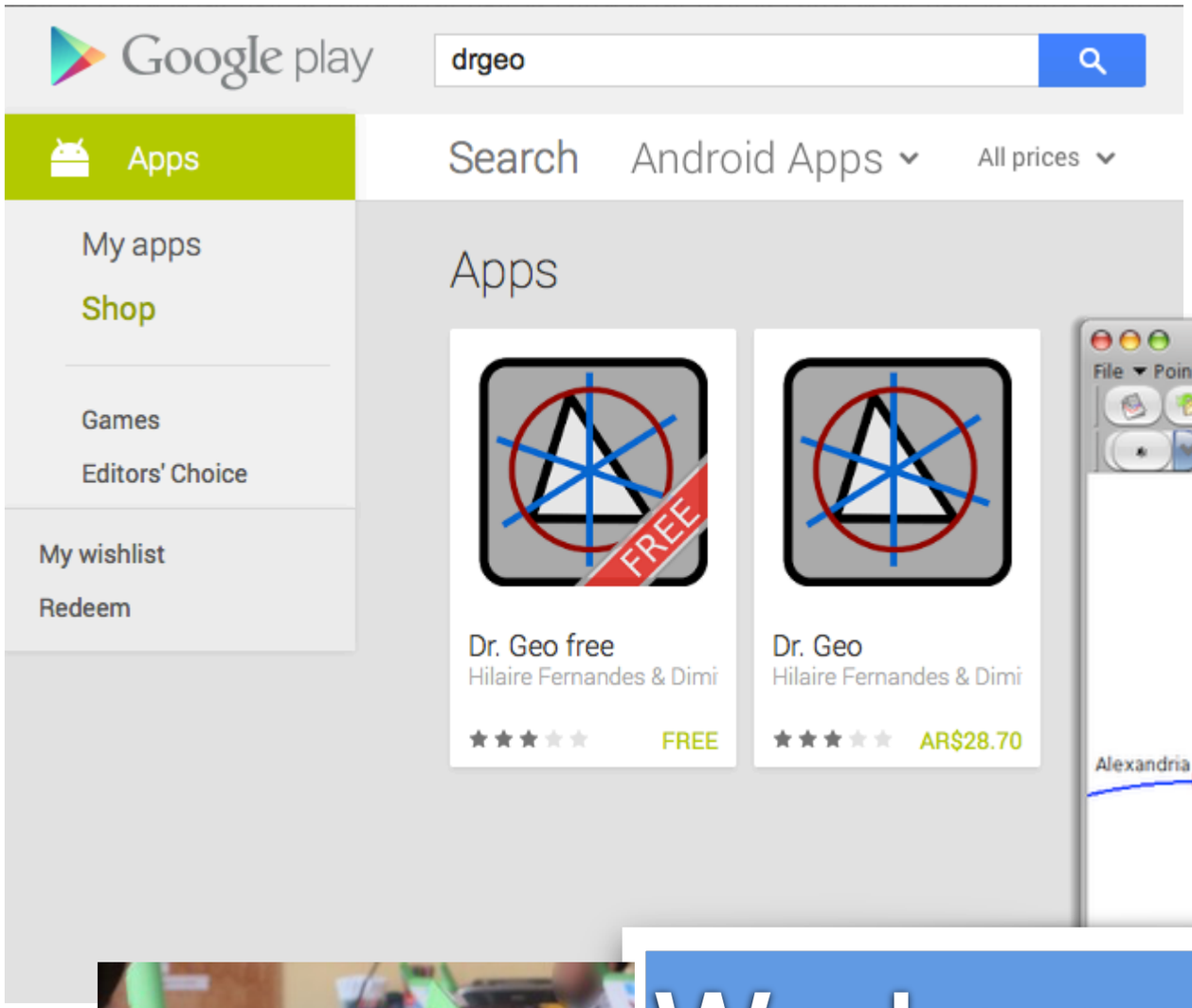
NTed: disaster scenario recovery

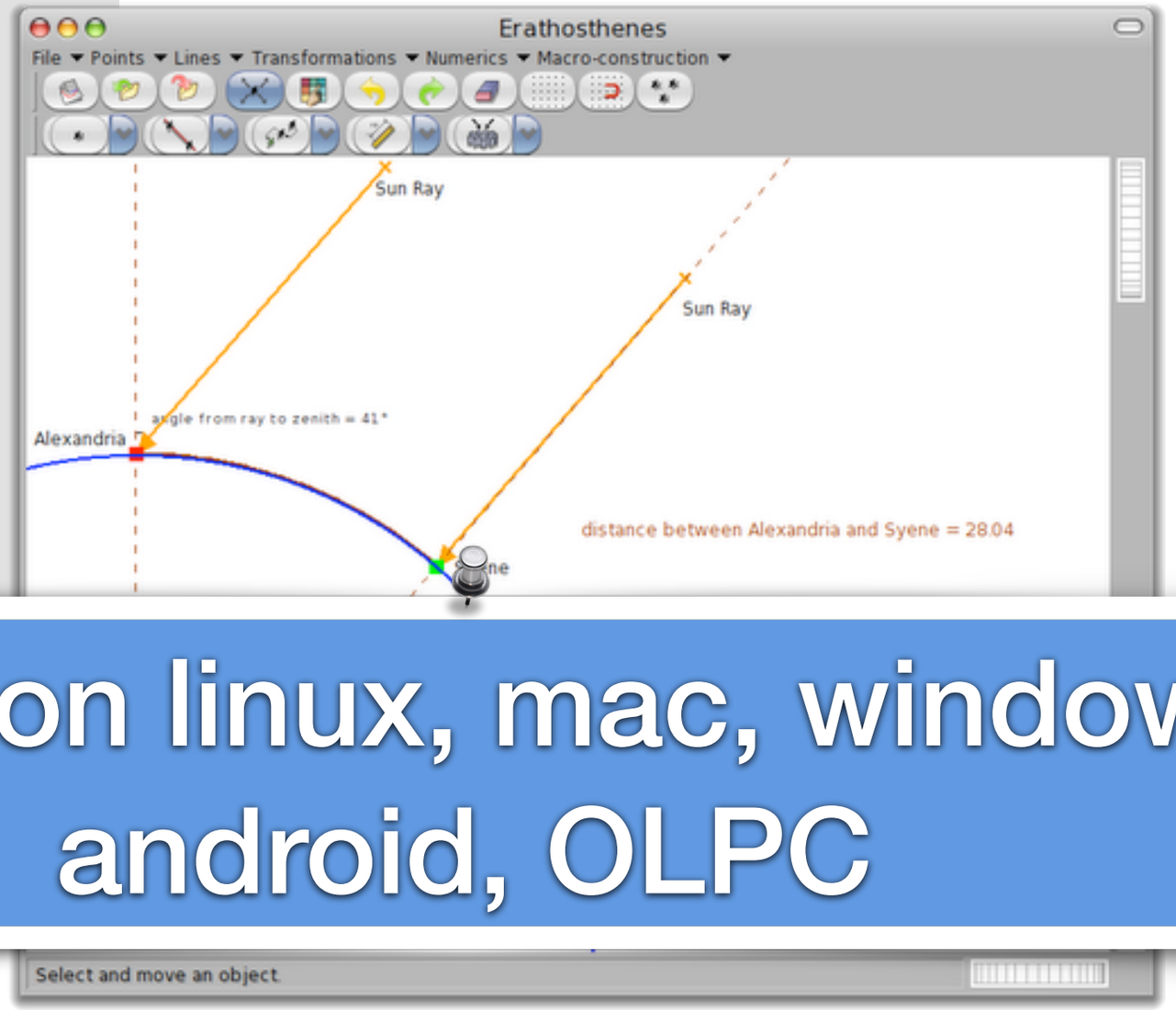


iBizLog - http://www.ibizlog.com

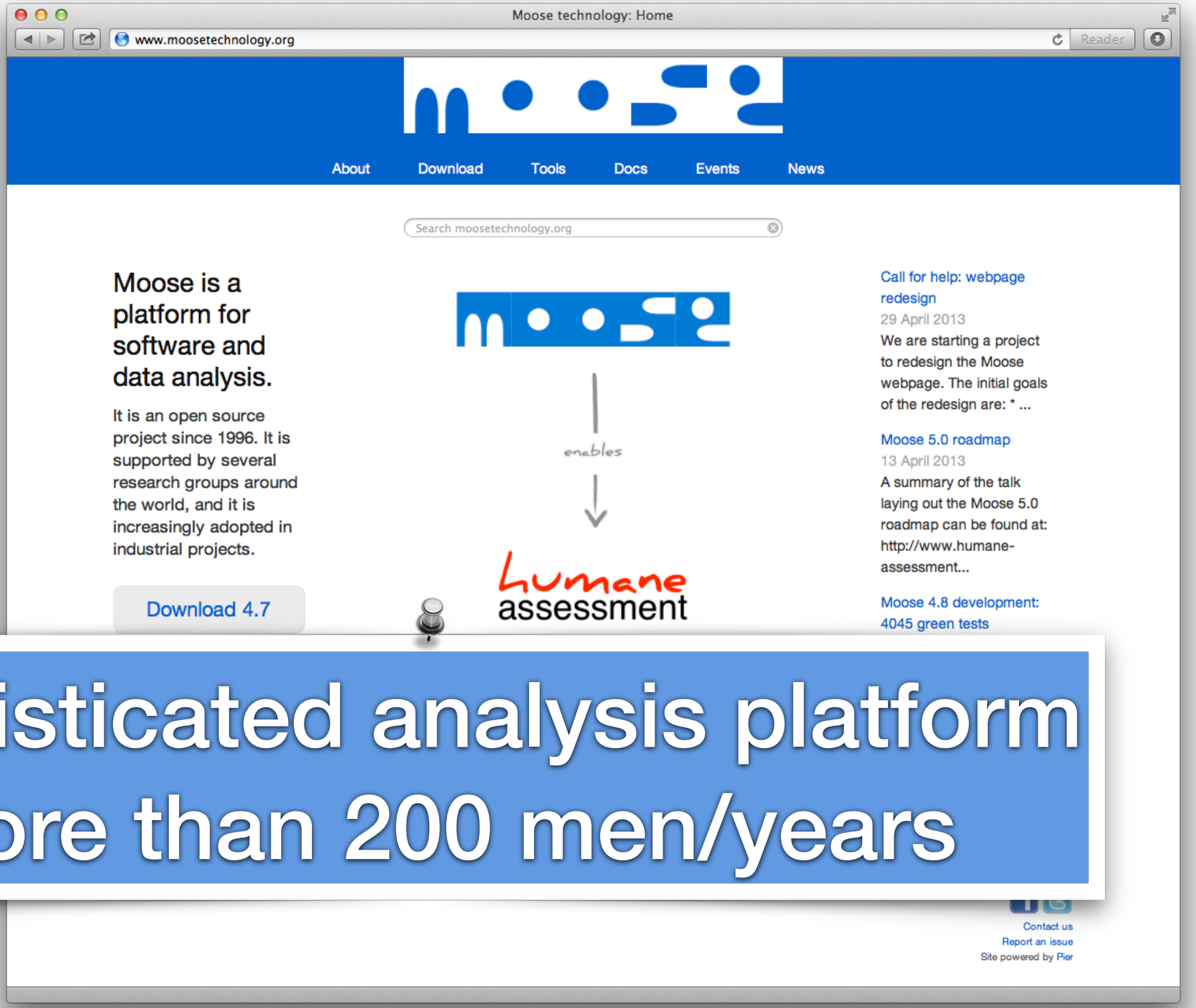
Full system develope
in 5 months one dev
(discontinued)

A product by Smallworks





Works on linux, mac, windows, android, OLPC



Sophisticated analysis platform more than 200 men/years

Pha

a TRMorph(1058537472)

zoom

a TRMorph(43778048)

ObjectProfile produces visualizations & analysis tools

SIGN IN | SIGN UP | LOCAL RESELLERS

NORIZZK-COM
Better Faster Cheaper Test Management

HOME | FEATURES | TRAINING & SUPPORT | PRICING

The world's first online platform fully supporting risk-based test management.

- Your IT is critical to your company.
- Would you dare not to control your risks and neglect the possible impact for your company and customers?
- Start controlling your risks in a standardized and easy way !!!

FREE TRIAL

1 BETTER

NORIZZK.COM guides you in composing the most effective and efficient end-to-end test strategy, starting from identified product risks. It helps you focus on the critical aspects of your projects and products.

NORIZZK.COM is the only platform for risk based test management showing you a comprehensive and visual overview on the risk status of your project.

YOUR ADDED VALUE

2 FASTER

NORIZZK.COM is the fastest way to compose your master and detailed test plan. It comes with a predefined knowledge base containing best practices for building your test plan, thus giving you a head start.

NORIZZK.COM allows sharing and reusing knowledge and experiences from previous projects. Over time, NORIZZK.COM becomes smarter in guiding you since it is self-learning and continuously improves its best practices.

YOUR ADDED VALUE

3 CHEAPER

As from now, wasting time and money by re-inventing the wheel is history. NORIZZK.COM gives you significant savings in creating your test plans.

It provides an intelligent and complete approach for all
sustai
maint

YO

COMPONENTS

Risk Assesment
Identify, categorize, prioritize and mitigate your risks with one single tool.

Test strategy
Write your master and detailed test plan in a faster way.

Test Estimations
Estimate all the required testing effort on one single page.

Test Results
Capture all the test results to monitor the progress.

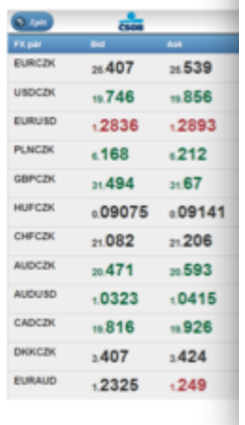
Reporting
Monitor the progress using the Risk

Export
Export your test plan to a docx file

Business Intelligence
Discover intelligence in your testing

LEARN MORE

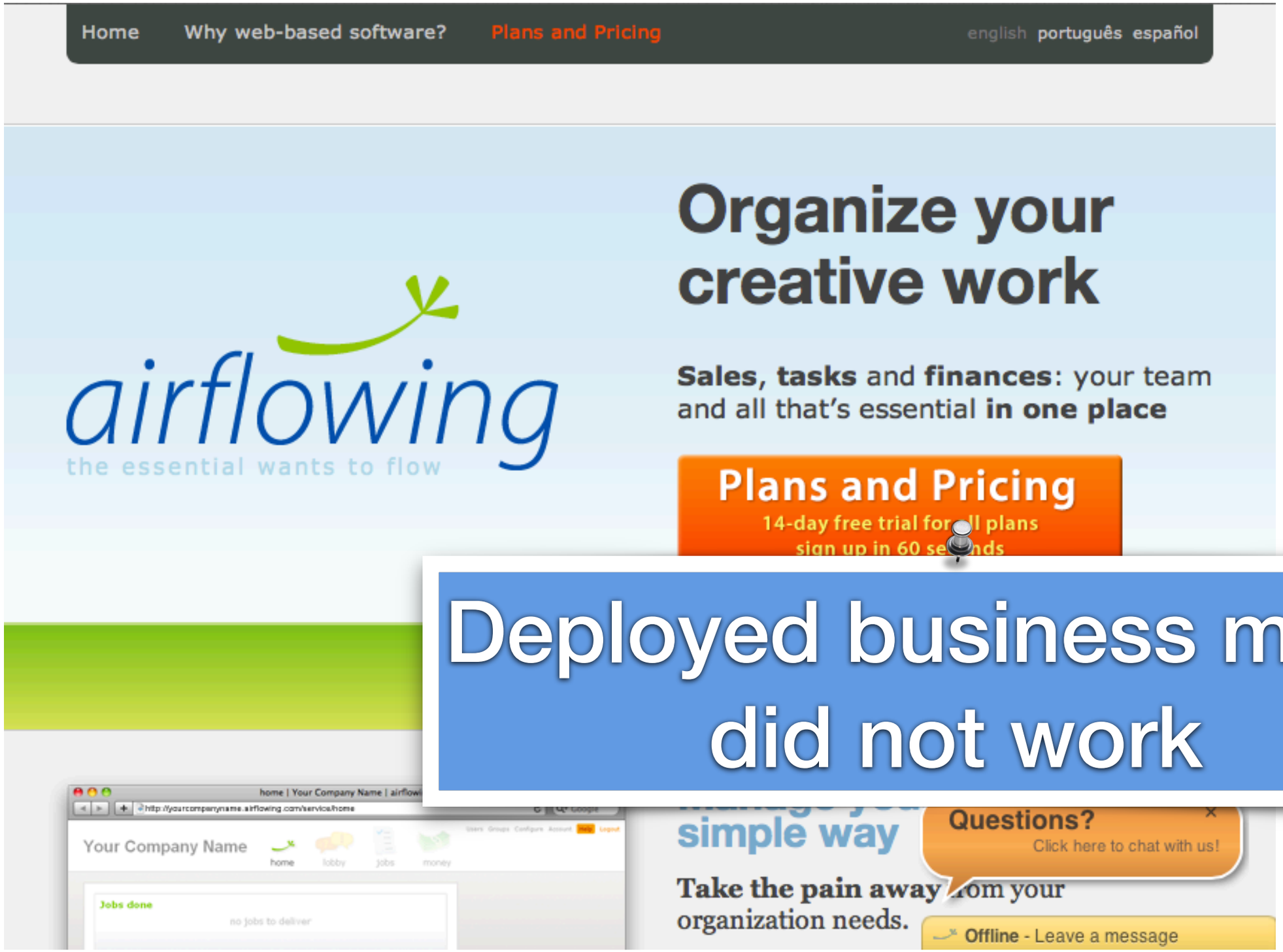
Digitalizing business



Developed in 3 months one dev

Sold to Belgium bank

Java team estimated 2 years


Home
Why web-based software?
Plans and Pricing
english português español
Organize your creative work
Sales, tasks and finances: your team and all that's essential in one place
Plans and Pricing
14-day free trial for all plans
sign up in 60 seconds
airflowing
the essential wants to flow
Deployed business model did not work
Your Company Name
home
lobby
jobs
money
Jobs done
no jobs to deliver
simple way
Take the pain away from your organization needs.
Questions?
Click here to chat with us!
Offline - Leave a message

synectique
Inventive Analysis

home products services contact

Can develop a complex analysis in 3 months

tools for software evolution

Dedicated Analyses

Synectique offers business intelligence tools for your

Dedicated tools

Synectique offers tools *fine-tuned to your software* analysis needs

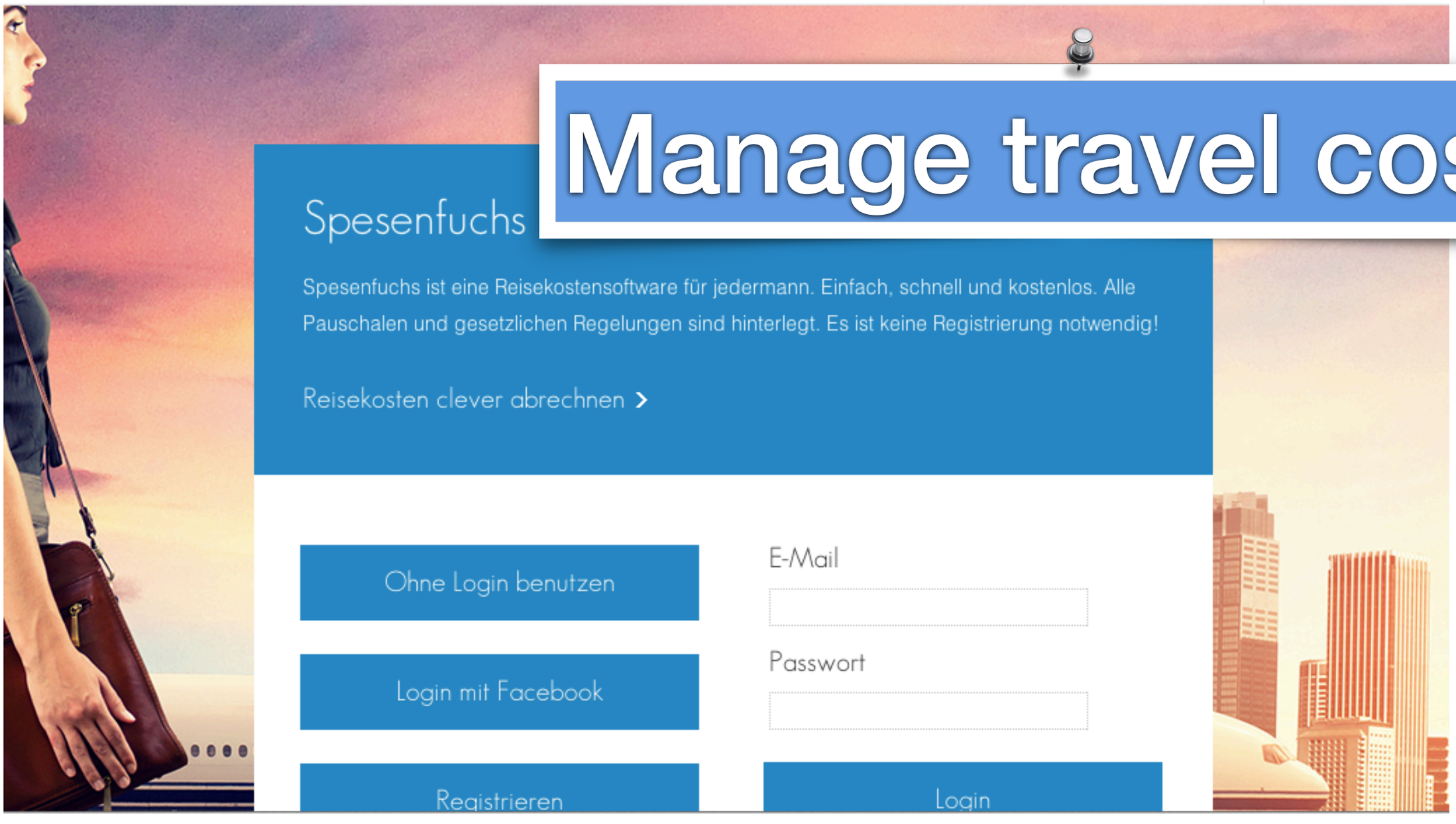
Decision making

Synectique tools provide *answers that lead to* concrete decisions

SpesenFuchs

Funktionen Aktuelles Forum FAQ Preise

Manage travel costs



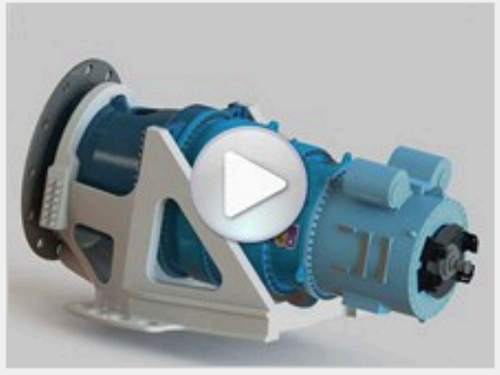
Wind Energy

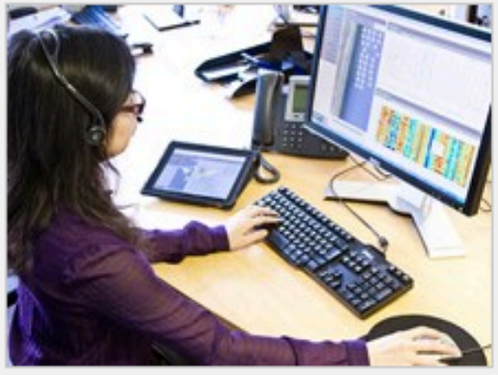
Pioneering new ways of maximising sustainable wind energy yields. Our products and services optimise asset availability, wind turbine performance and drivetrain reliability. We work with owners, operators, manufacturers, insurers and service providers worldwide.

Get in touch

We provide services and software to a range of industries

ENQUIRE TODAY ›





Related links

InSight
RomaxWIND
Butterfly platform
Forensic engineering
Analysis service
Monitoring service

Overview | Why we are specialists | Who we work with

Wind Energy

Tweet Share +1 SHARE

Pioneering new ways of maximising sustainable wind energy yields

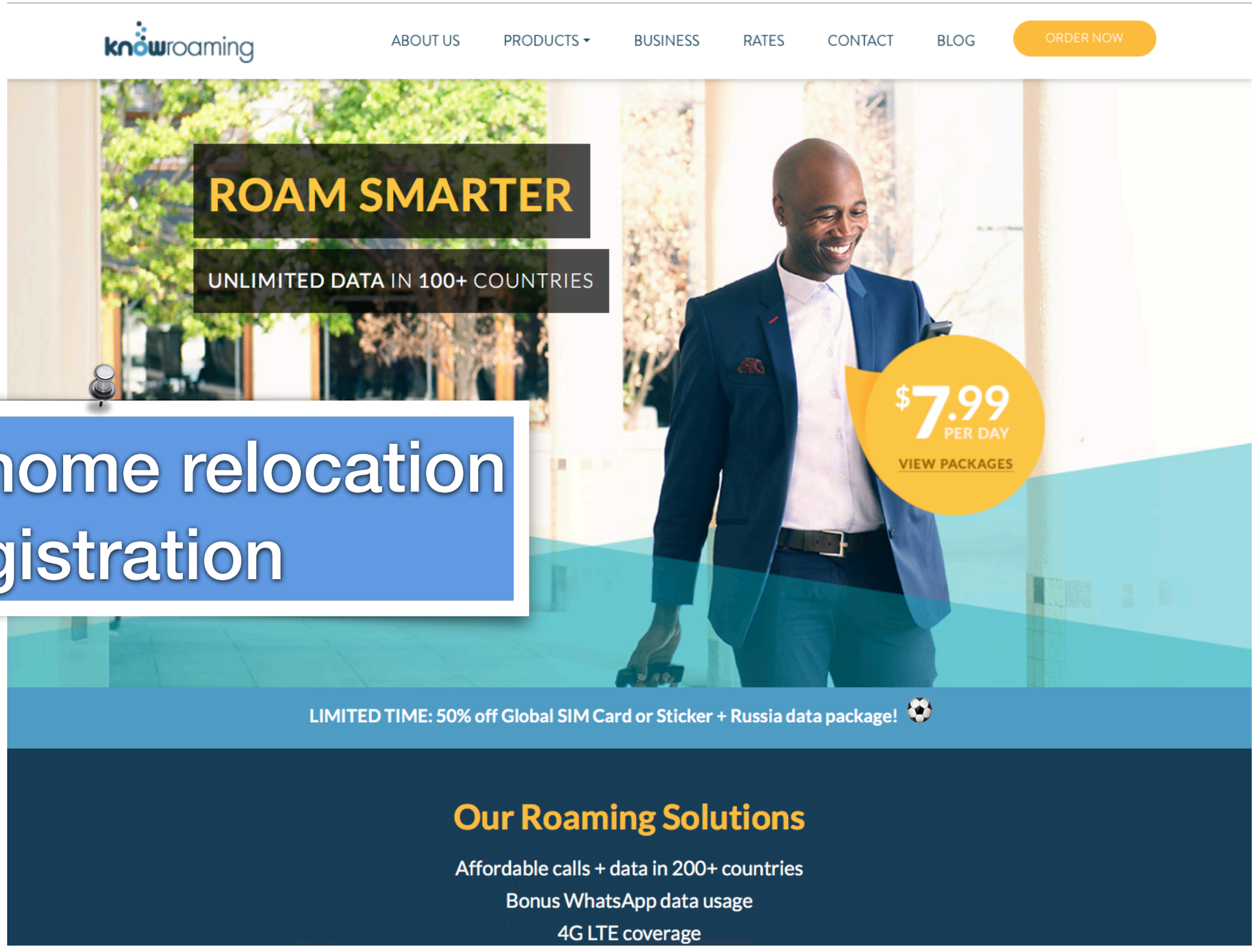
Wind industry solutions

scan share send

Handle return mails in Switzerland

Built a dataflow plug and play on top of Pharo

Specific home relocation registration



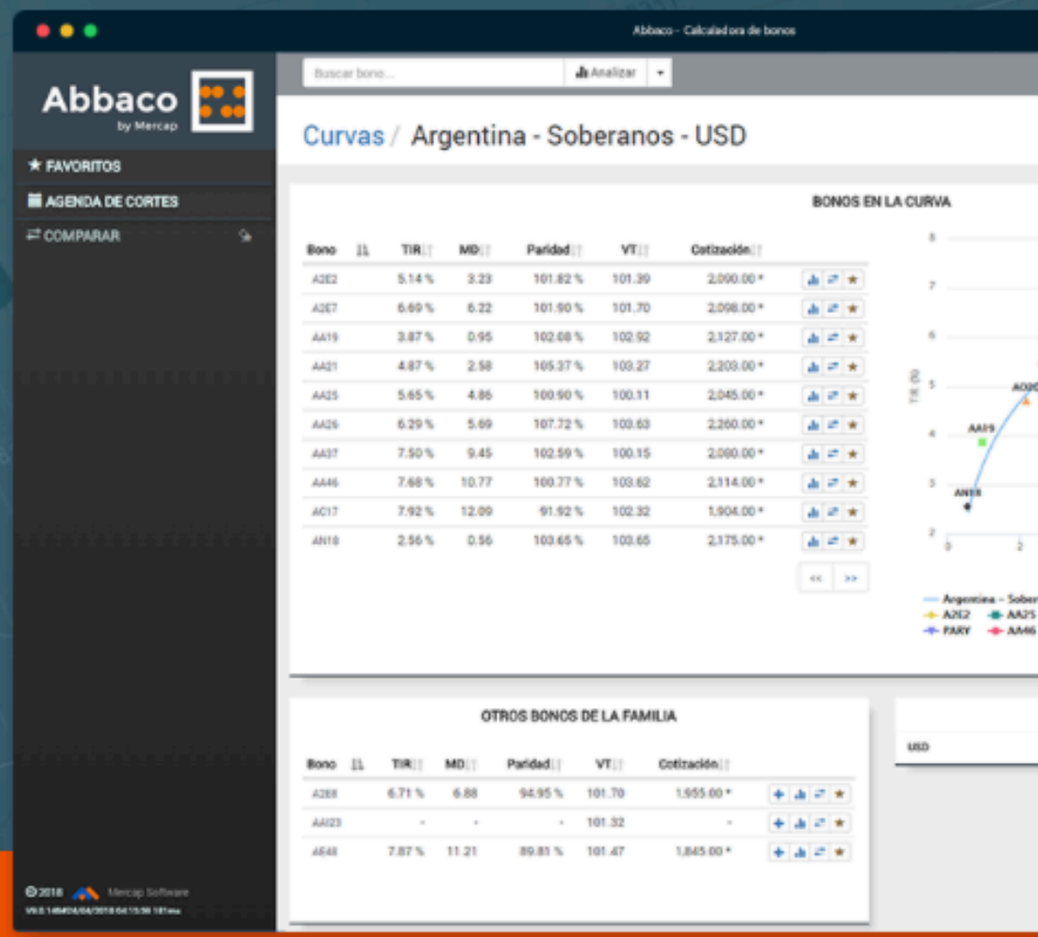
Abbaco by Mercap

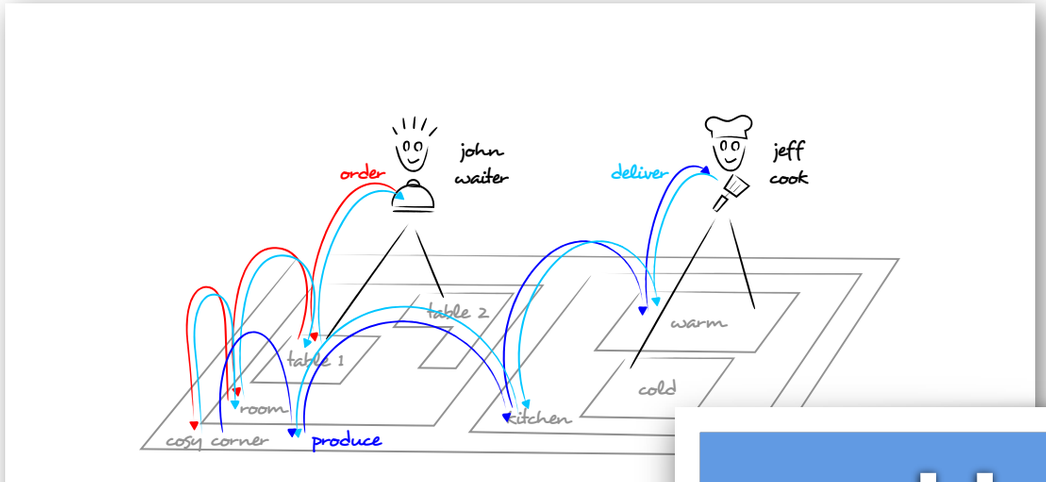
Features Advantages Pricing Customers **Try for free** Login ES

Bond calculator for experts investors

Your own research tool for making intelligent decisions.

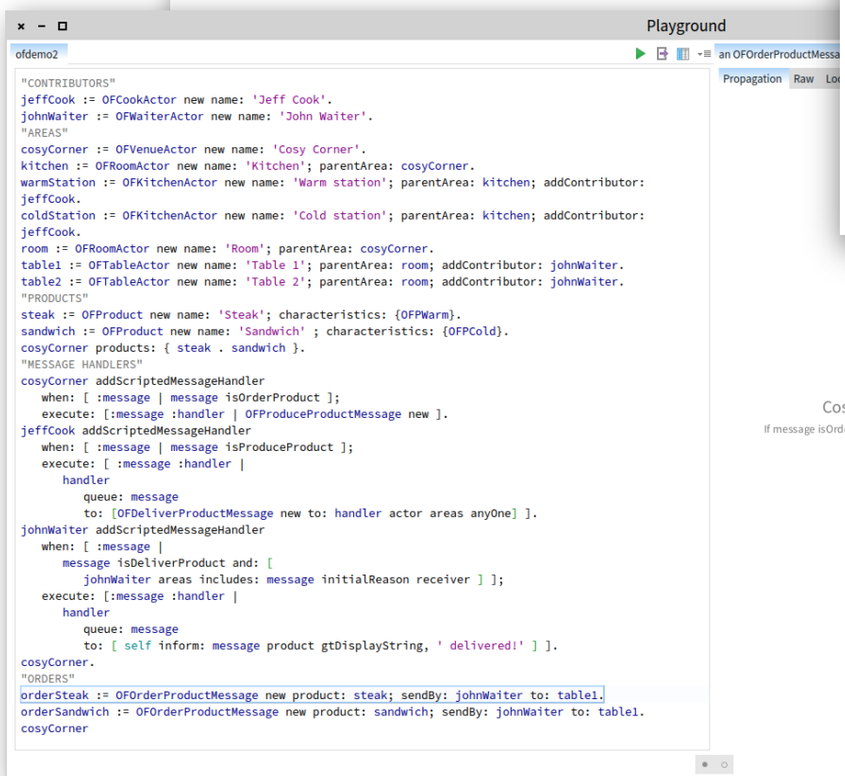
from Argentina

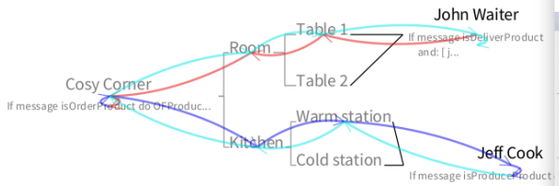


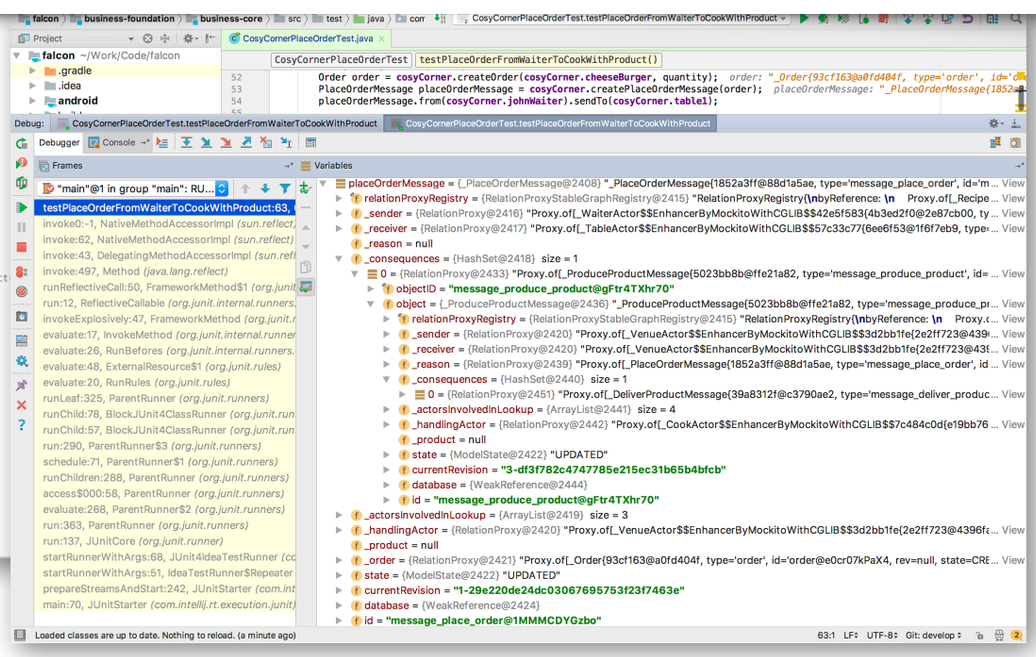


feenk

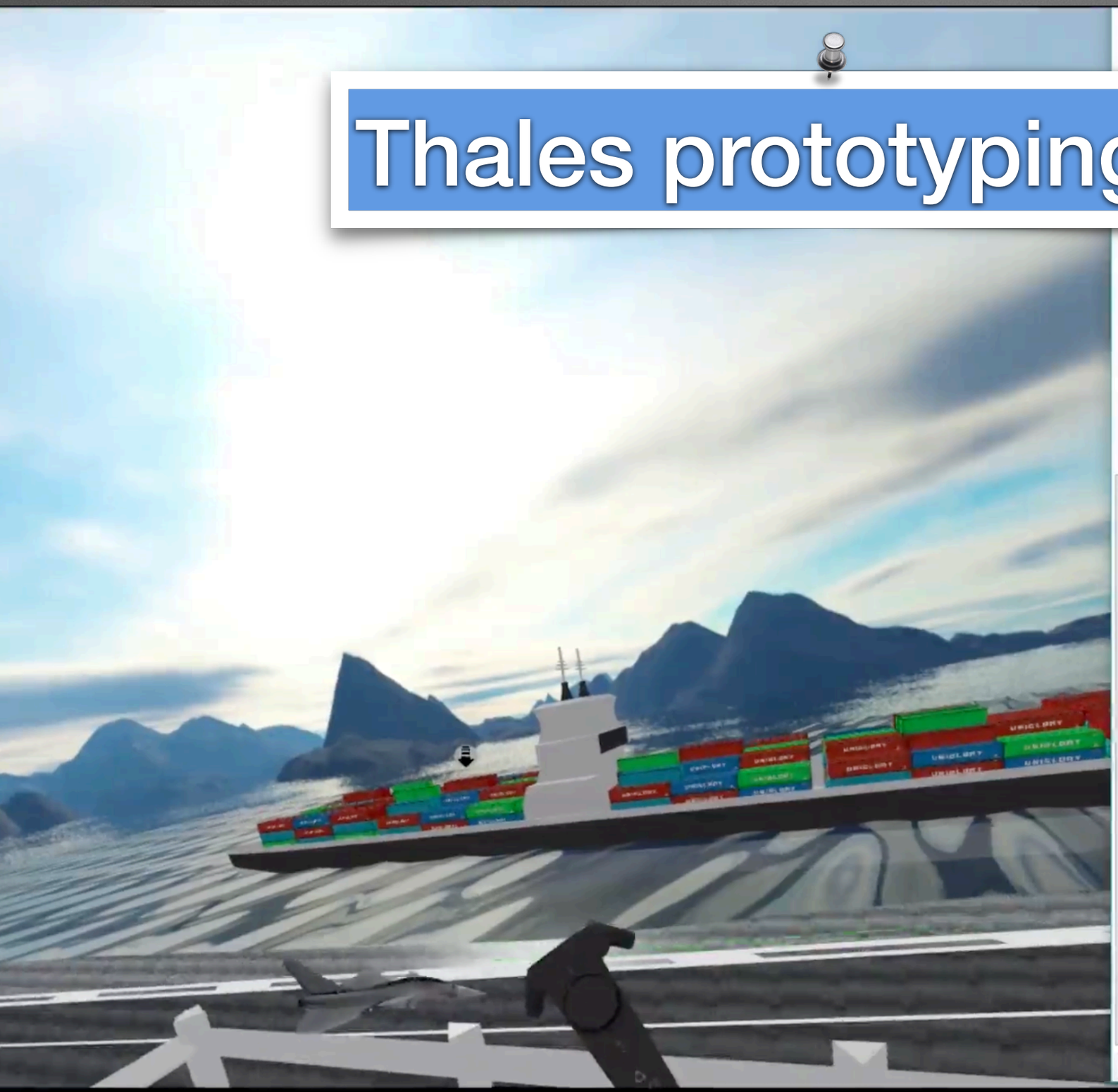
Help growing startups
Get back control on softwa







Thales prototyping VR



Playground

raftCarrier open.

tmp

OVRWDemoAircraftCarrier

Scoped | Variables

History Navigator

Type: Pkg1|*Pkg2|Pk...

OpenVR
OpenVR-Woden
OpenVR-Woden-
Pharo-Help
Polymorph-Task
Polymorph-Widg
PragmaCollector
ProfStef-Core
ProfStef-Help
ProfStef-Tests

ProtoObject
Object
WDApplication
WDFPSExample
OVRWEmptyExample
OVRWExample
OVRWDemoAircra
OVRWDemoPickir
OVRWExampleCol

Hier. | Class | Com.

-- all --
event handling
initialize
menu
scene content
updating

addAbeilleBourbon
addBarrel
addBoxes
addCargo
addCharlesDeGaulle
addHawkeye
addHelicopter
addObjectAt:orientation:s
addPlayableAreaToScene
addRadar

```
OVRWExample subclass:
#OVRWDemoAircraftCarrier
   instanceVariableNames:
'flyingRafaleNode teleportLocations
scrollGestureListener menu
operationQueue waterMaterial
dynamicsWorld'
   classVariableNames: ''
   package:
```

1/11 [2] -- 1/11 [25] Format as you read W +L

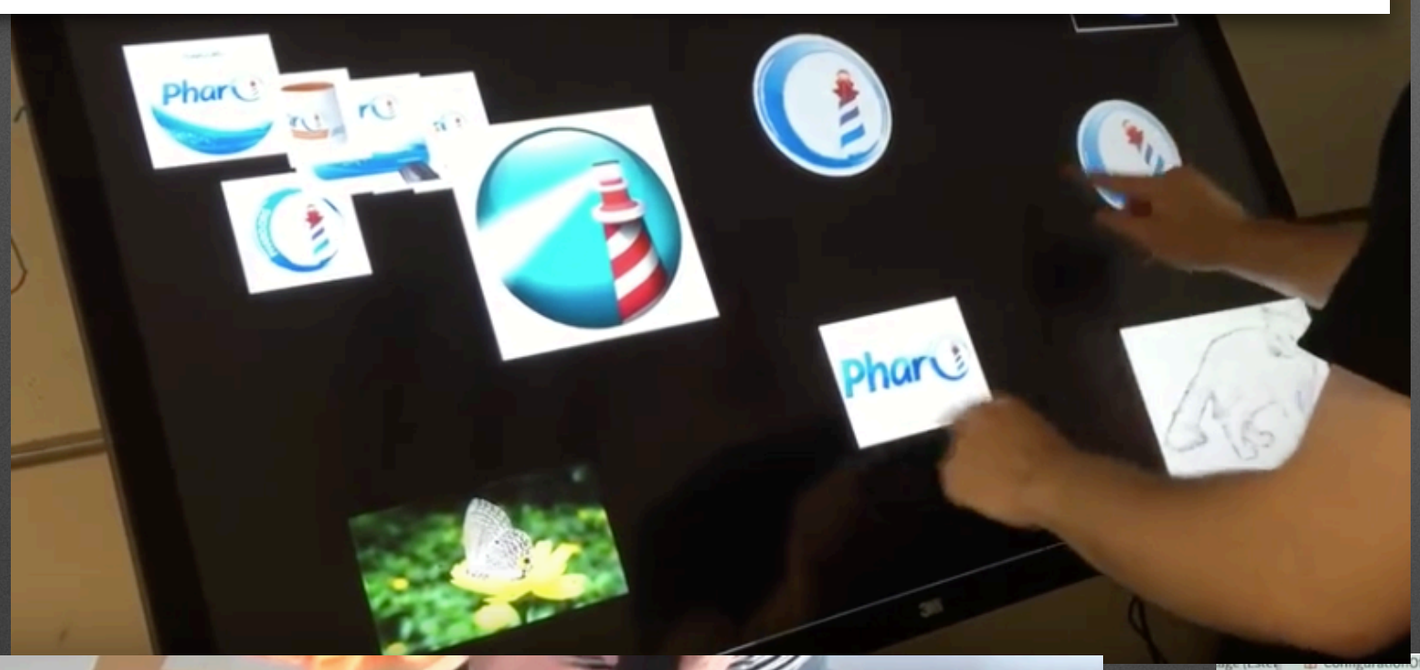
Controls:
- Big front button for teleportation
- Menu button to change the color of the sea

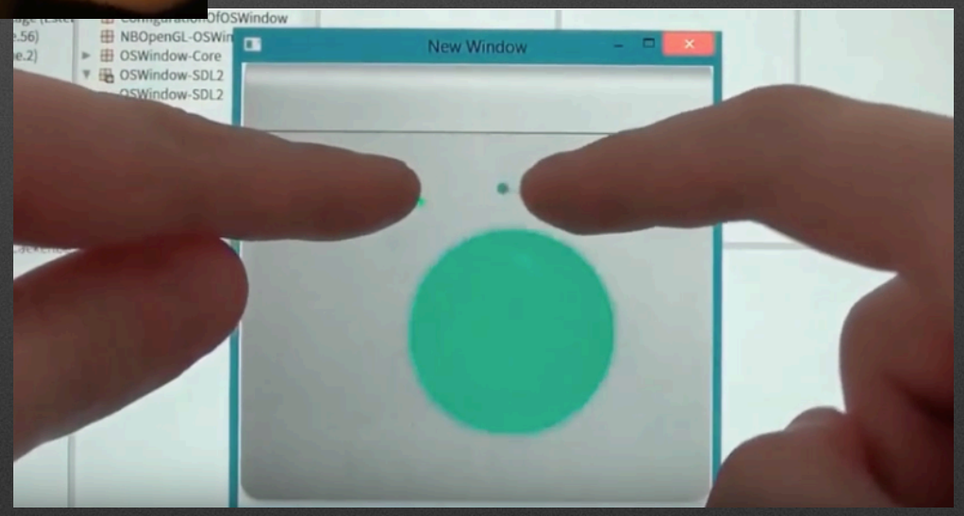
1/3 [1] W +L

Class not referenced ? Helpful?
[dynamicsWorld] Instance variables not read AND written ? Helpful?
[dynamicsWorld] Variables not referenced ? Helpful?

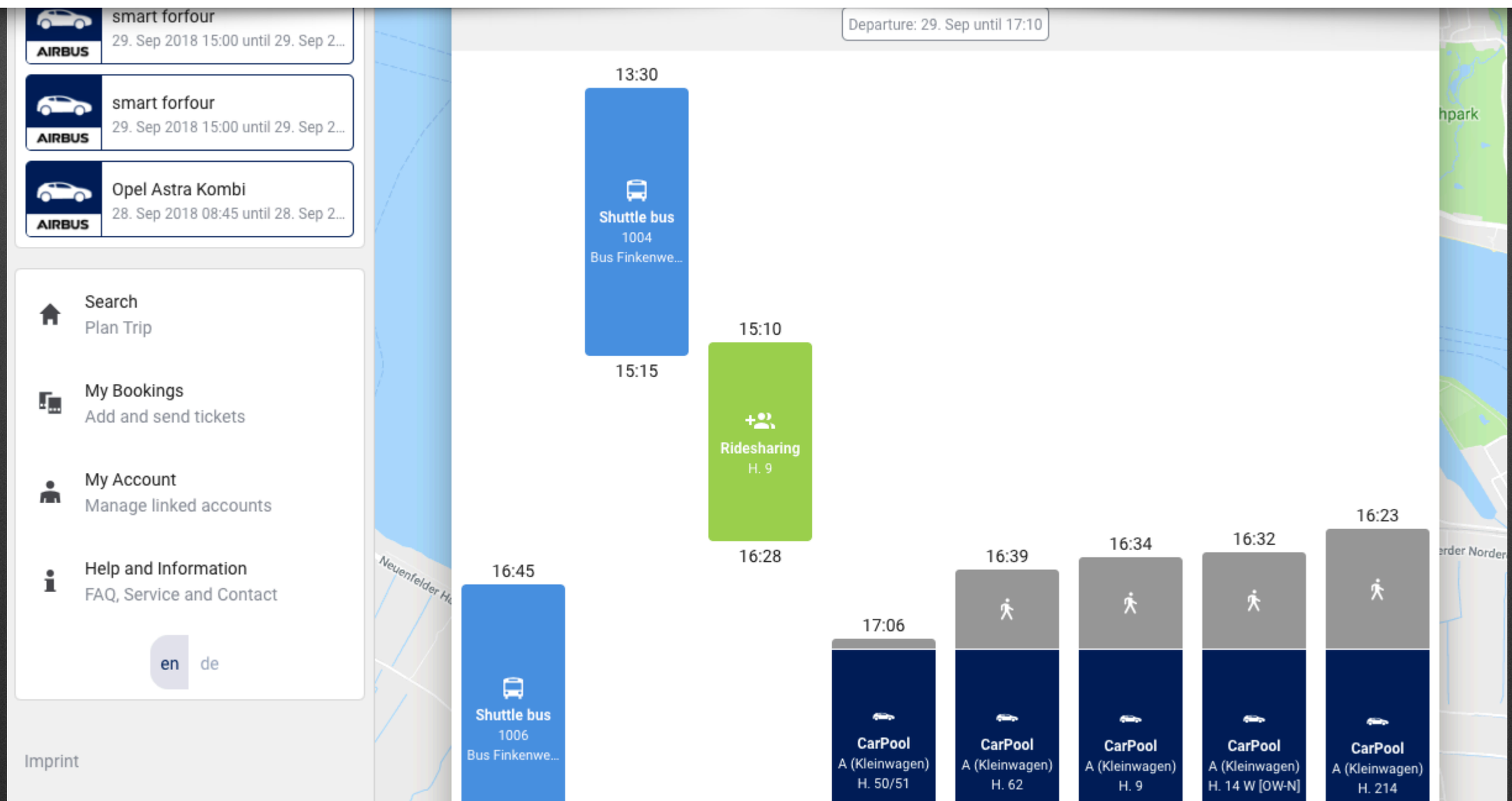
OVRWDemoAircraftCarrier | Playground

Thales prototypes advanced interaction design

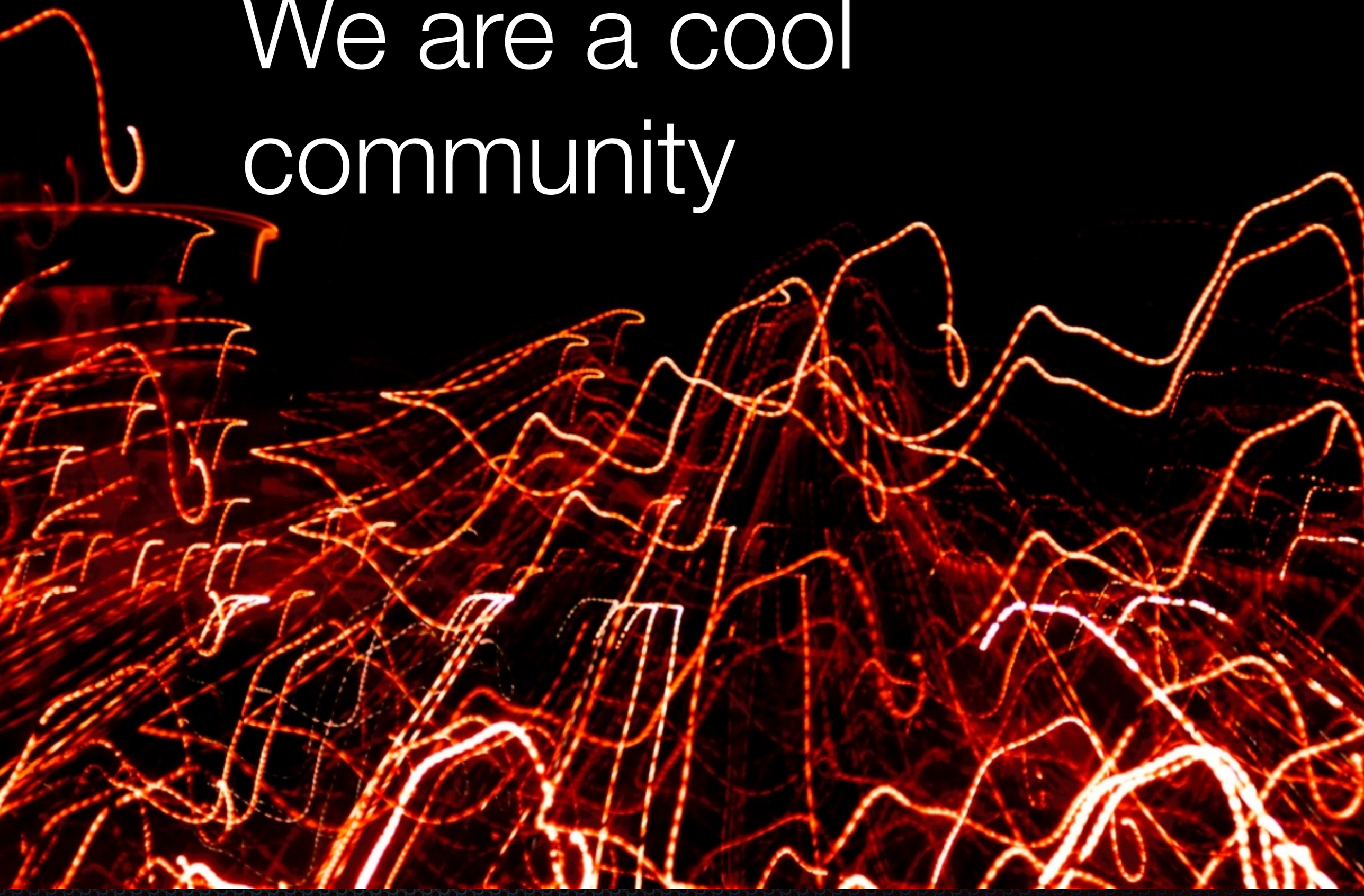




Mobility Map: a broker for mobility services
multi-modal routing search
docker running dozens of pharo



We are a cool community



Some Pharo's Teachers

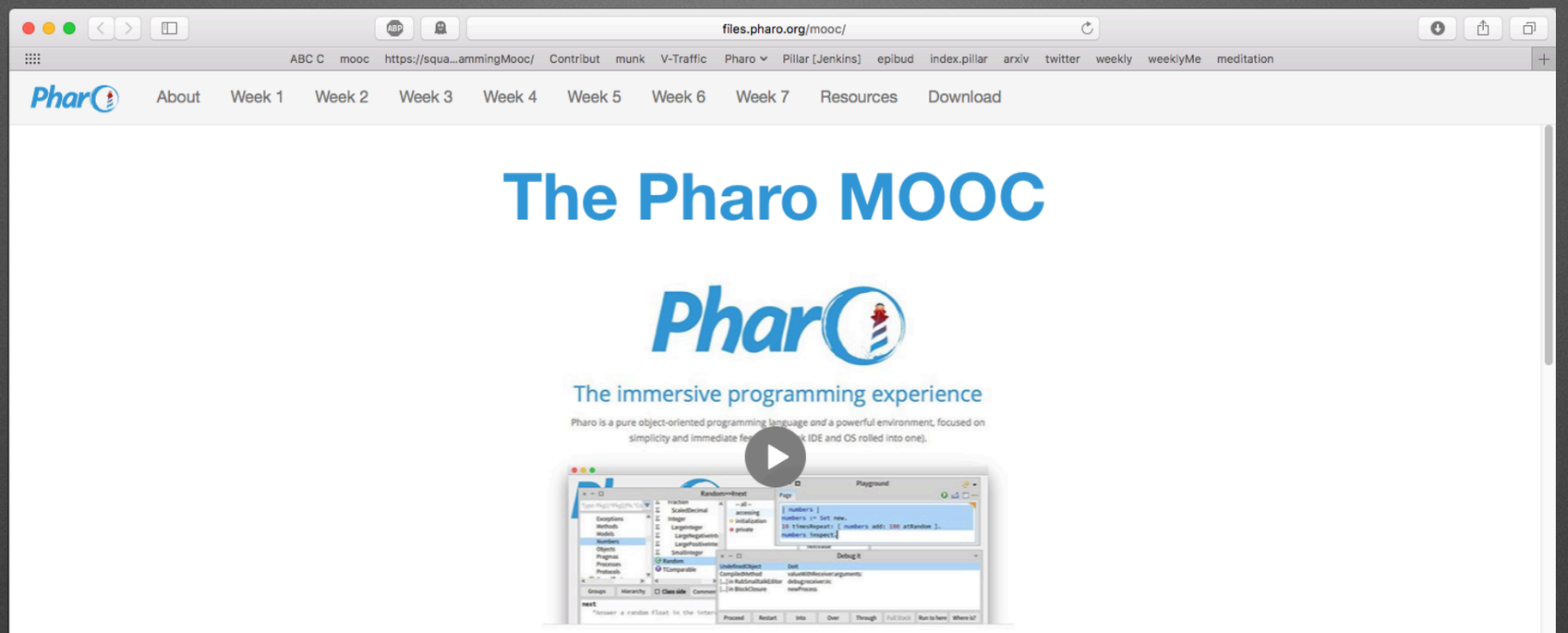
• Univ. of NoviSad • Uni. of Buenos Aires • Uni. of Bern • Uni. of Maroua • Uni. of Brussels • Ecole des Mines de Douai • Uni. de Savoie • Ivan Franko Nat. Uni. • Czech Technical Uni. • CULS Prague • Uni. of Quilmes • Uni. of La Plata • Northern Michigan Uni. • Uni. Technologica Nacional (UTN) • Uni. 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. of Yaounde • Uni. Lille • INFTI • Uni. of Lyon •

Some Research Groups

Lafhis (AR)
CAR (FR)
RMOD (FR)
Ummisco (IRD)
Reveal (CH)
Lysic (FR)
CEA-List (FR)
Uqbar (AR)
OC (FR)
CCMI-FIT (CZ)

ASERG (BR)
Pleiad (CL)
SoftQual(Serb)
SRA (Jap)
GL (Bol)
UFMG (Br)
ETS (Can)
U.Chi (Can)
Soft (Be)
…

http://mooc.pharo.org



About this course

If you are either a beginner or an expert in object-oriented programming, this MOOC will change the way you program with objects: come and learn or rediscover object-oriented programming with Pharo!

Pharo is a pure object-oriented programming language in the tradition of Smalltalk. It offers a unique developing experience in constant interaction with live objects. Pharo is elegant, fun to use and very powerful. It is very easy to learn and enables to understand advanced concept in a natural way. When programming in Pharo, you are immersed in a world of live objects. You have immediate feedback at any moment of your development on objects representing web applications, code itself, graphics, network. More…

Obtaining the MOOC

We strongly suggest to follow the MOOC during the next session on France Université Numérique that will start **October 16th, 2017** because you will have support from the teaching team.

Join the discord channel mooc on the Pharo community (Invitation at http://discord.gg/Sj2rhxn)

But, you can also access all ressources of this MOOC offline on:

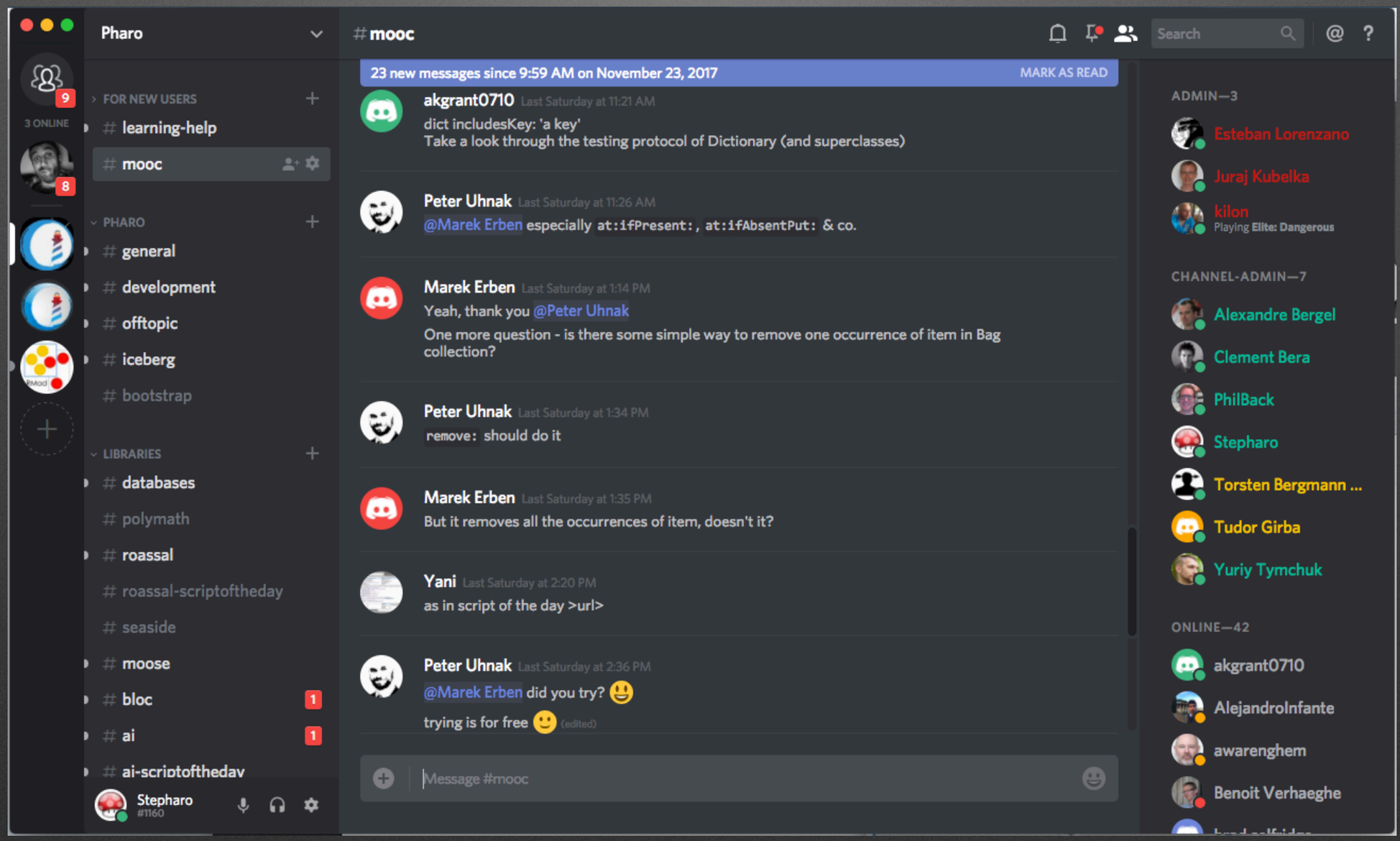
- WebPortal: full online lecture with all videos and exercises,
- CanalU: one of the key website that delivers ressources for teaching and research.

How to follow this MOOC?

We propose five profiles to follow this MOOC at different speed and levels. The Mooc sequences will be tagged according to these profiles. The profiles are not exclusive: to become a pharo expert you need to start as a newbie and grow up. Of course, you can also follow the full MOOC.

- **Total newbies [Newbies]** Focus on the beginning of the lectures, you can skip the Seaside/Mongo and tinyBlog exercises. Follow the end of the Mooc but without trying to understand everything. Take it easy and spend time understanding that the system is written in itself. Focus on the navigation tools (senders and implementors).
- **Looking for Web [Web]** For those that focus on web development, we suggest to get started with the syntax (since we really love this lecture you should really watch the not implementation

http://discord.gg/Sj2rhxn



http://books.pharo.org

Pharo Books

Pharo is a clean, innovative, open-source, live-programming environment.

Contribute to the SquareBracketsAssociates community free books
https://github.com/SquareBracketAssociates/

Access the book compilation farm!

Pharo by Example, intended for both students and developers, will guide you gently through the Pharo language and environment by means of a series of examples and exercises.

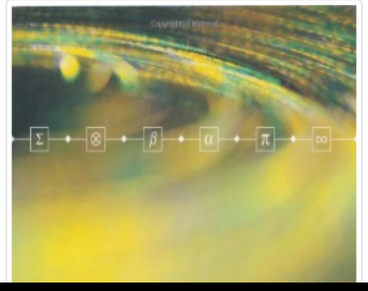
View details »

Deep into Pharo is the second volume of a series of books covering Pharo. Whereas the first volume is intended for newcomers, this second volume covers deeper topics.

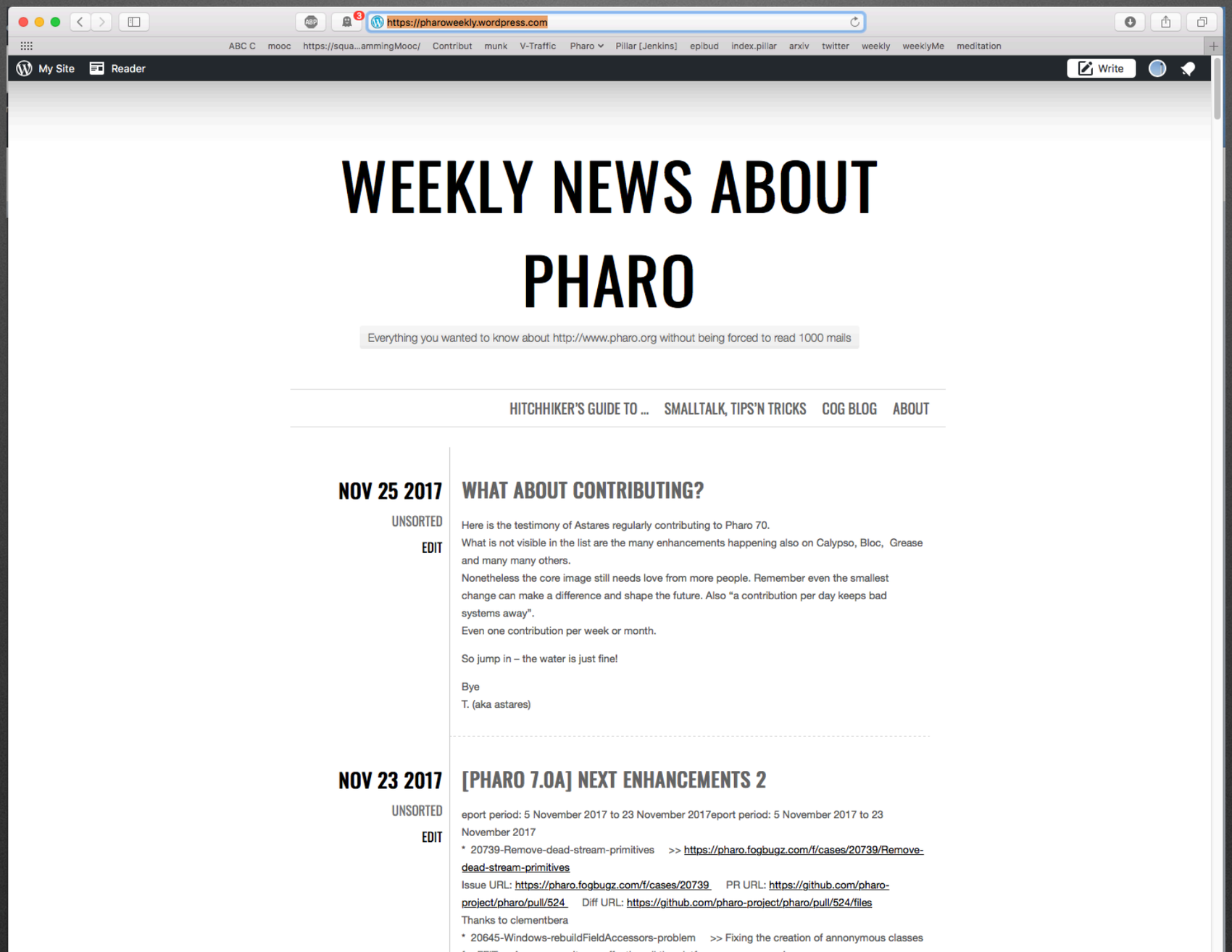
View details »

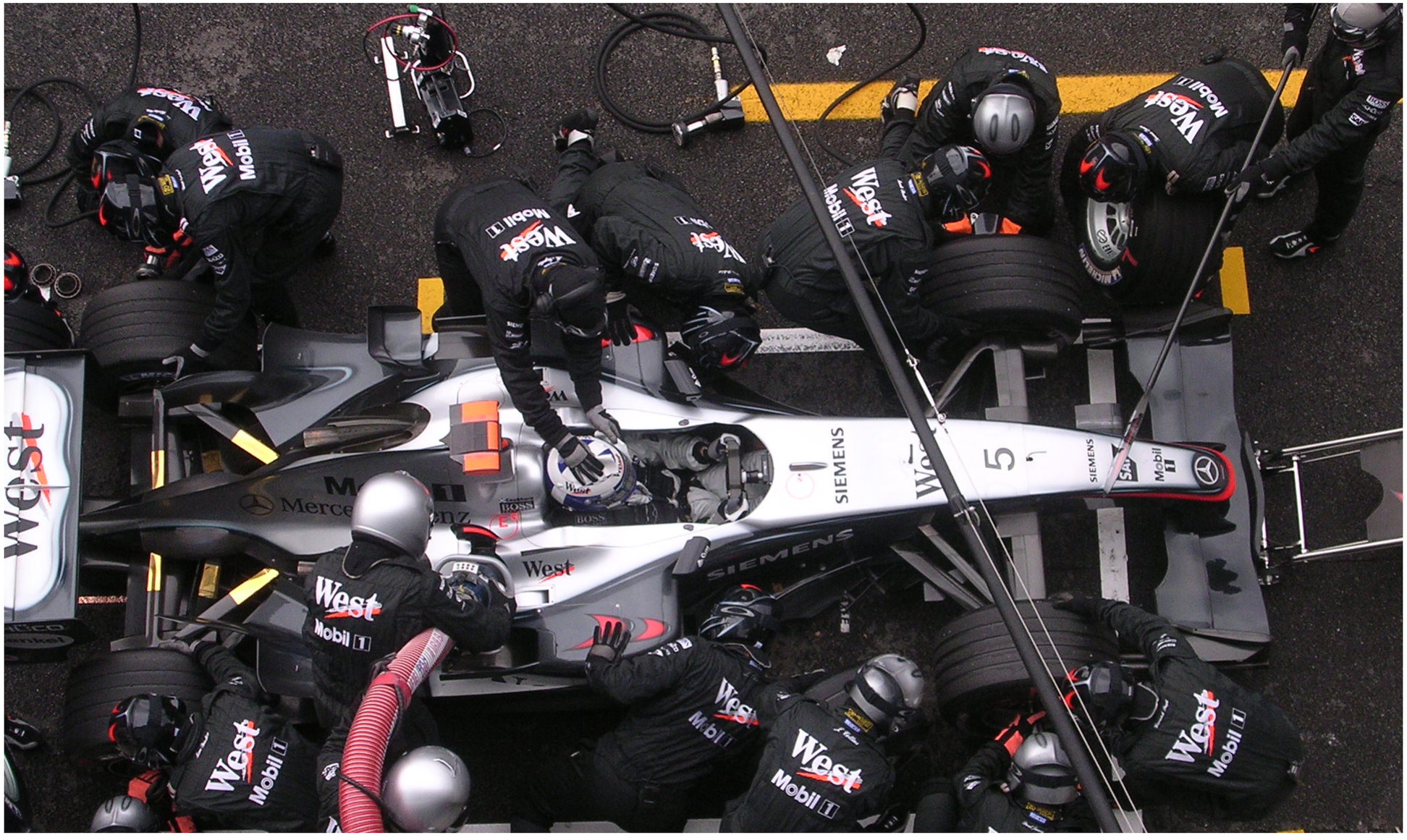
Enterprise Pharo is the third volume of the series. This book covers libraries and frameworks for enterprises, and in particular those doing web development.

View details »



https://
pharoweekly.wordpress.com





Pharo is our vehicle
We improve it everyday

For us Pharo is the beginning of the journey and we are building the road

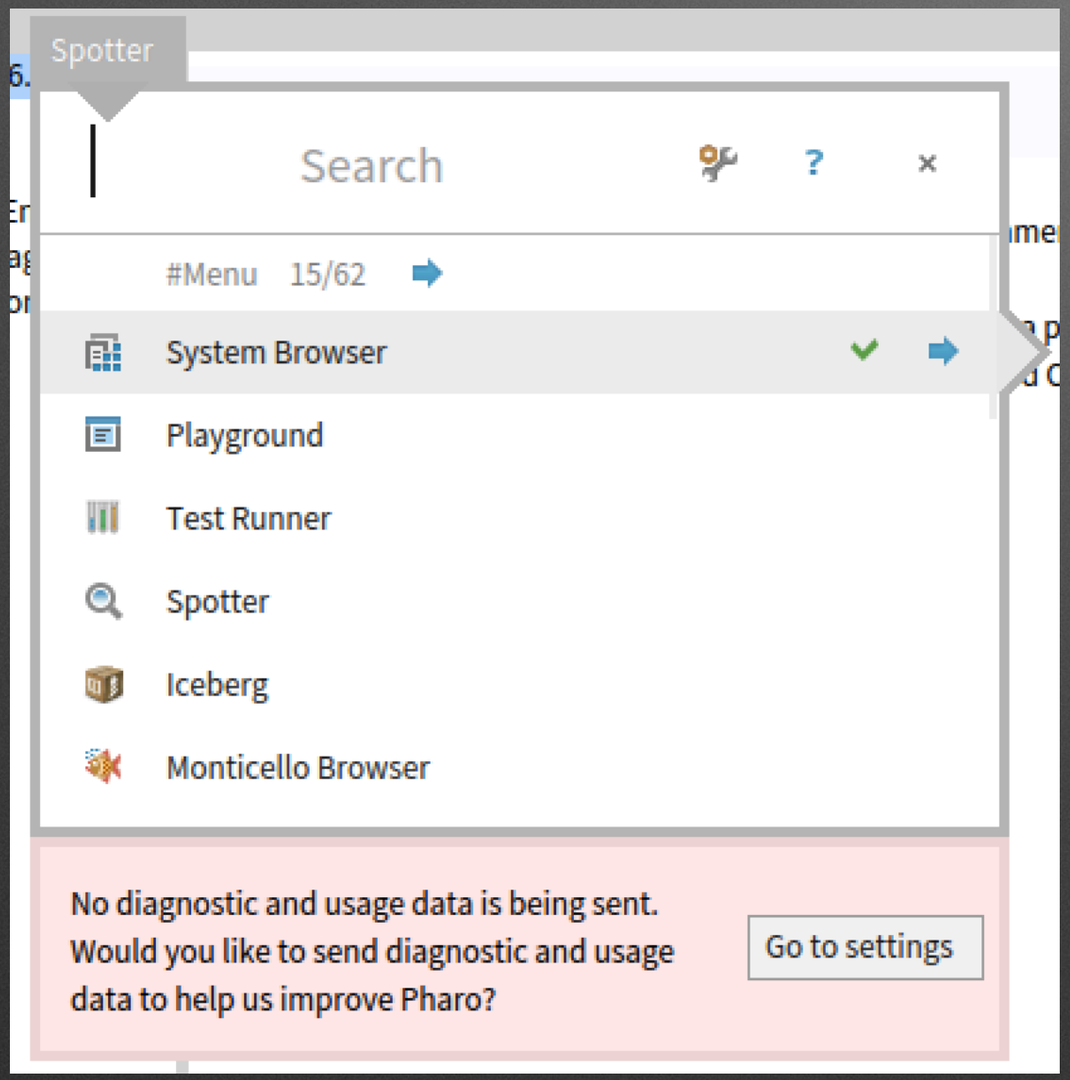
Pharo is

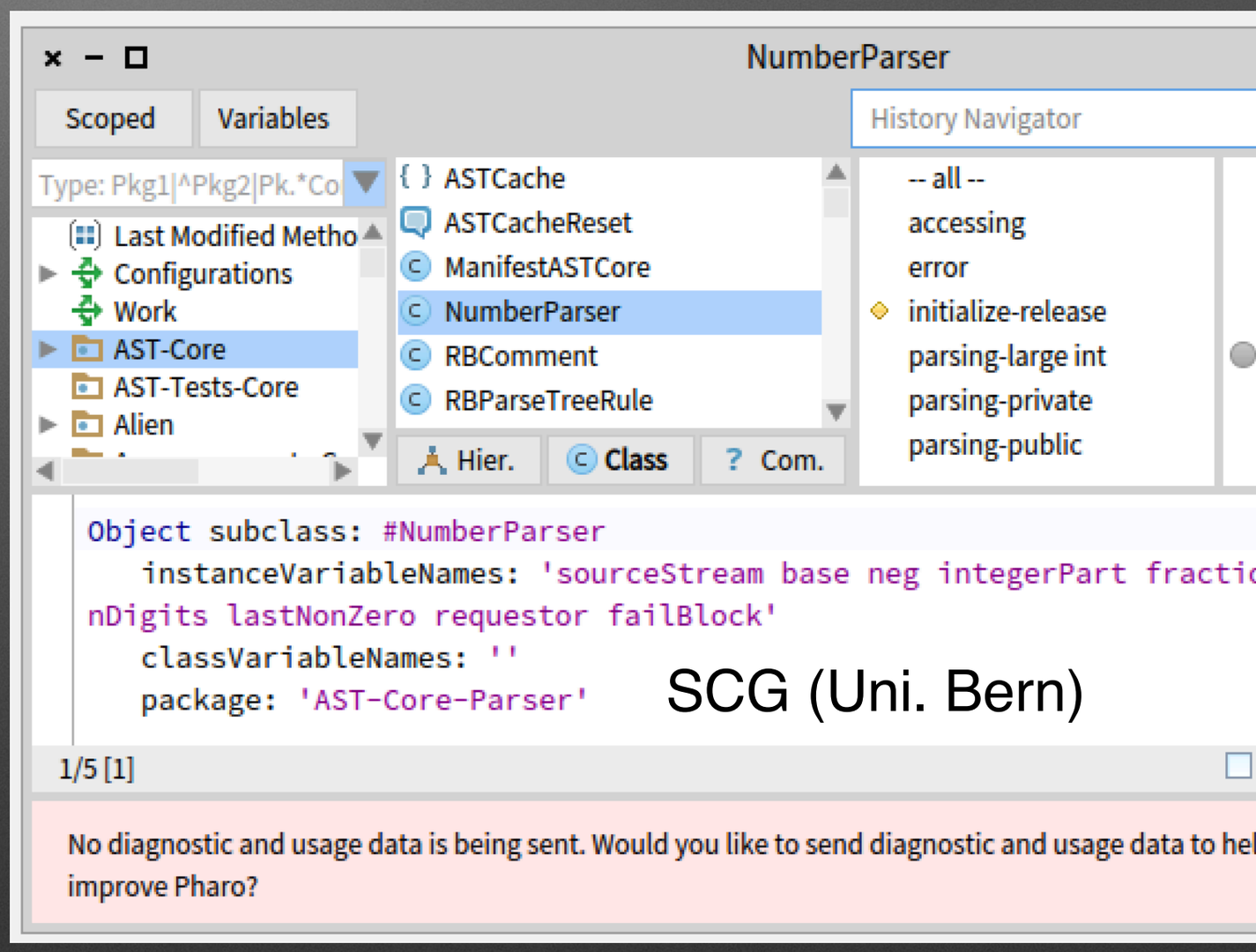
research friendly

We are ready to help you validate your ideas

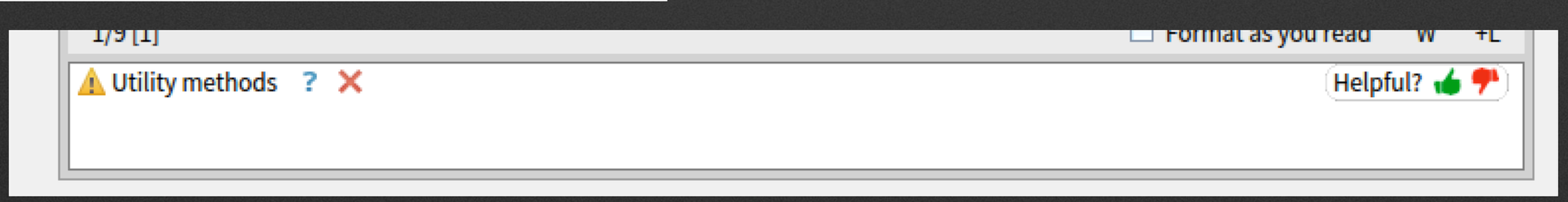
We can be guinea pigs

Actively supporting research





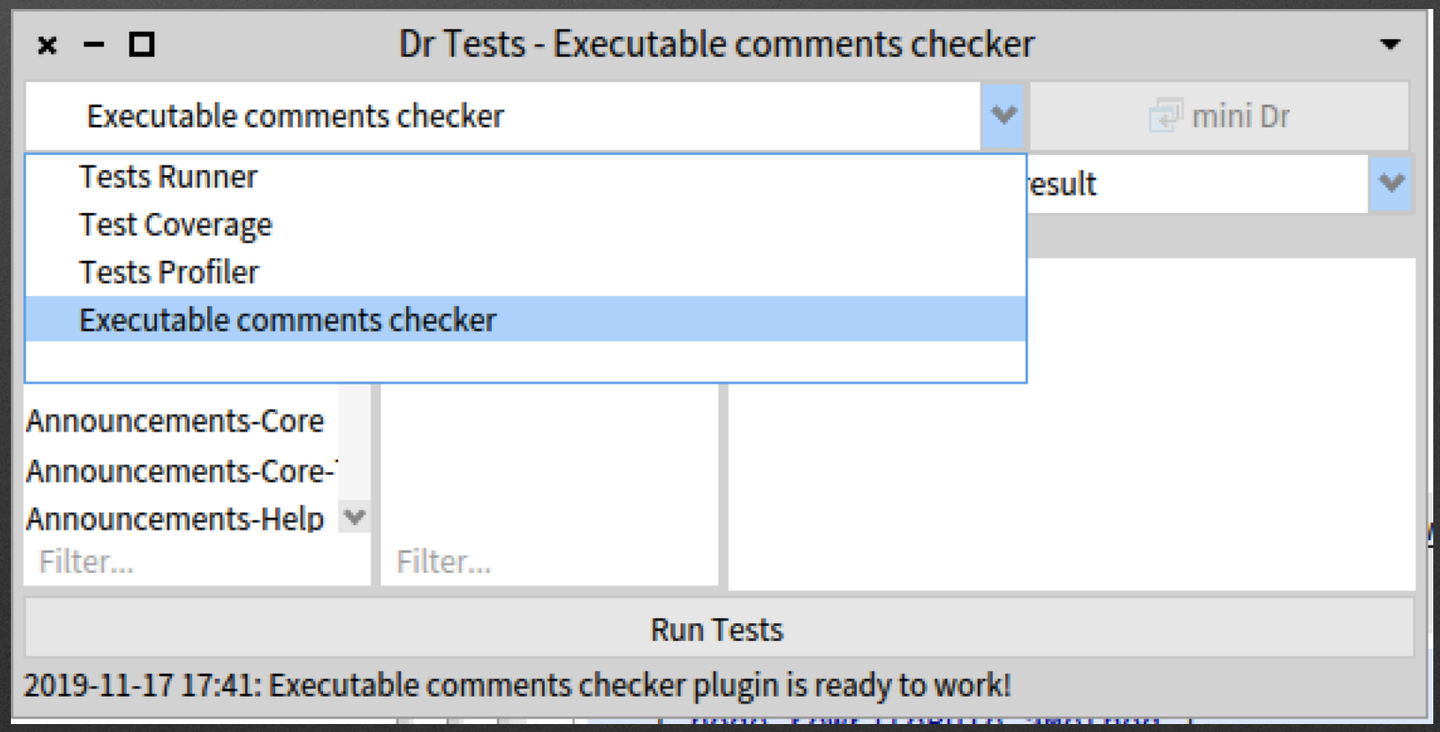
SCG (Uni. Bern)



Best Paper Award @ ICPC'19

Kubelka, Bergel, Robbes,
"Live Programming and Software Evolution:
Questions during a Programming Change Task"

DrTests: a plugin-based architecture to plug test analyses...



We validated our 25000 tests for Rotten Green Tests (ICSE'19)

Test Amplification

by S. Demeyer, H. Rocha, M. Abdi

Code Review

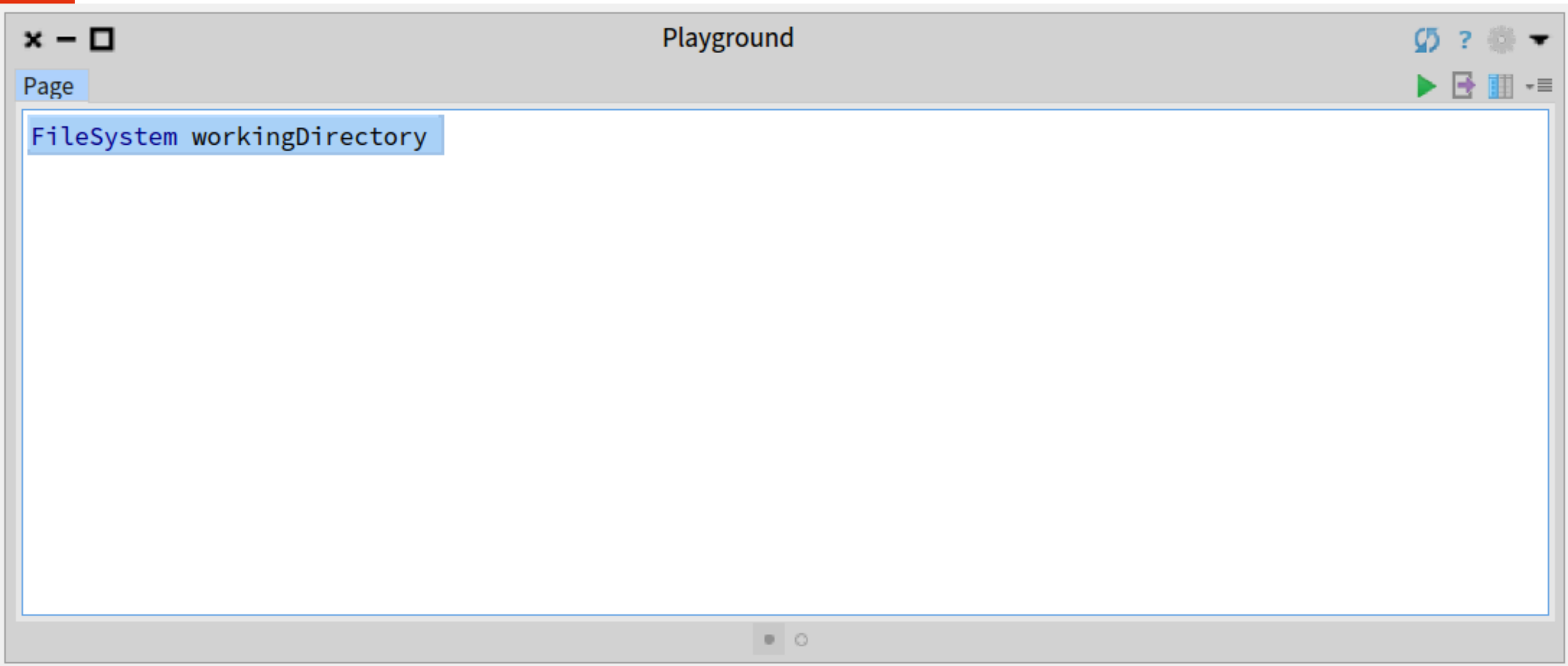
by A. Bachelli, ObjectProfile

What about adapting tools to our problem instead of the inverse?

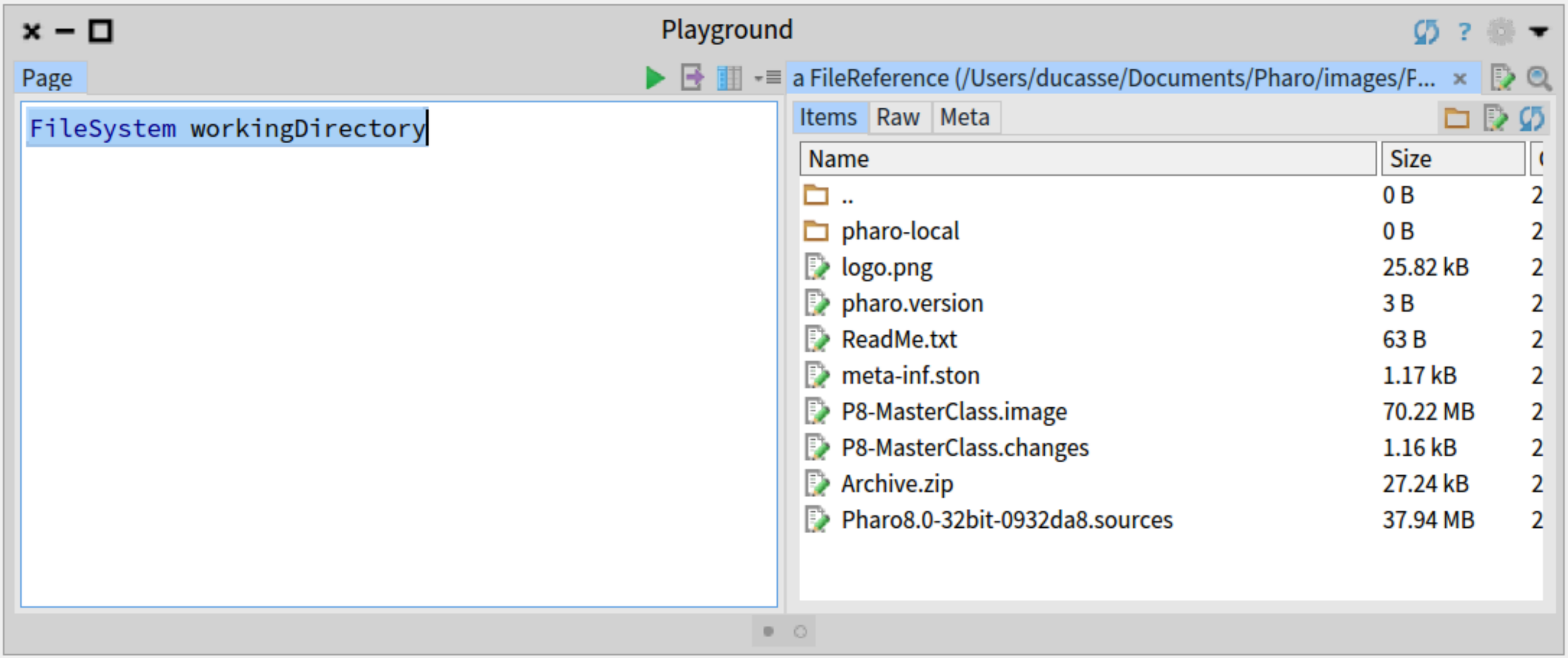
Not so dull example!

The views of a file reference

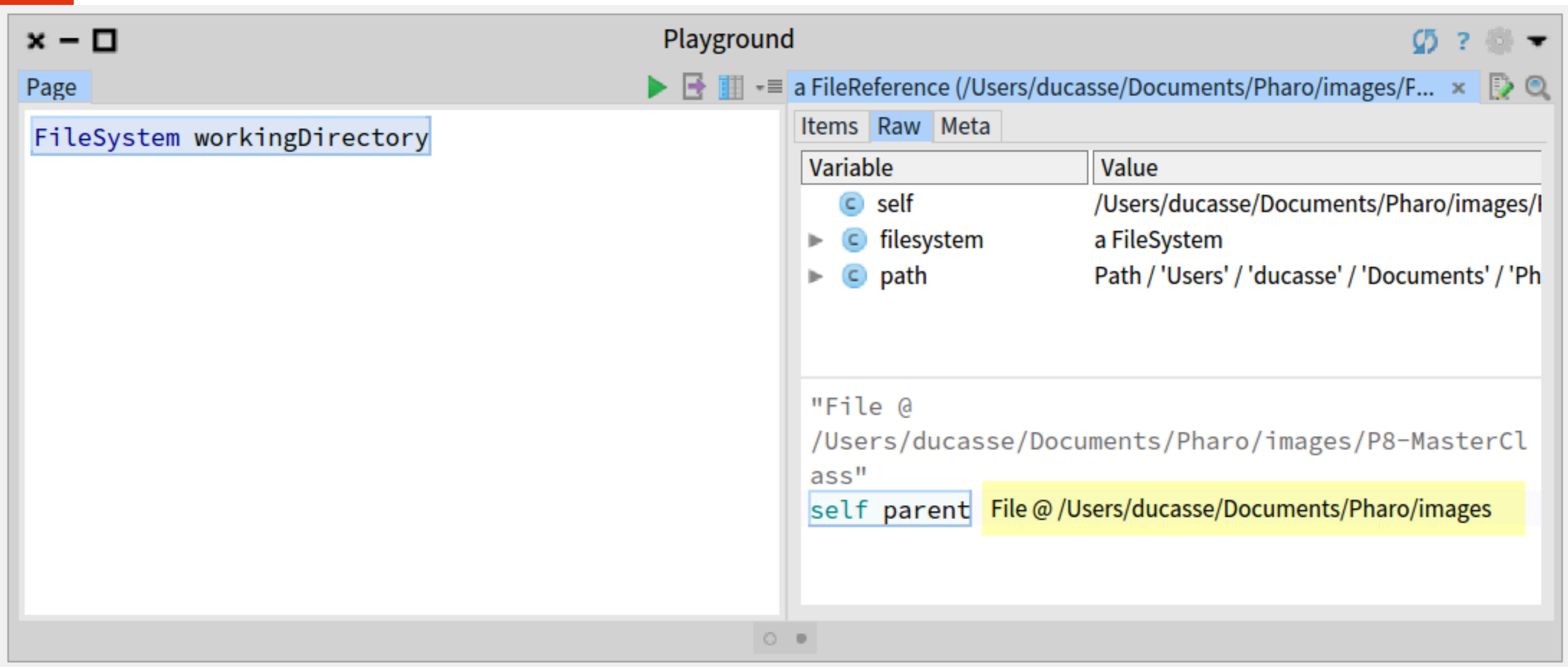
Looking at a file reference



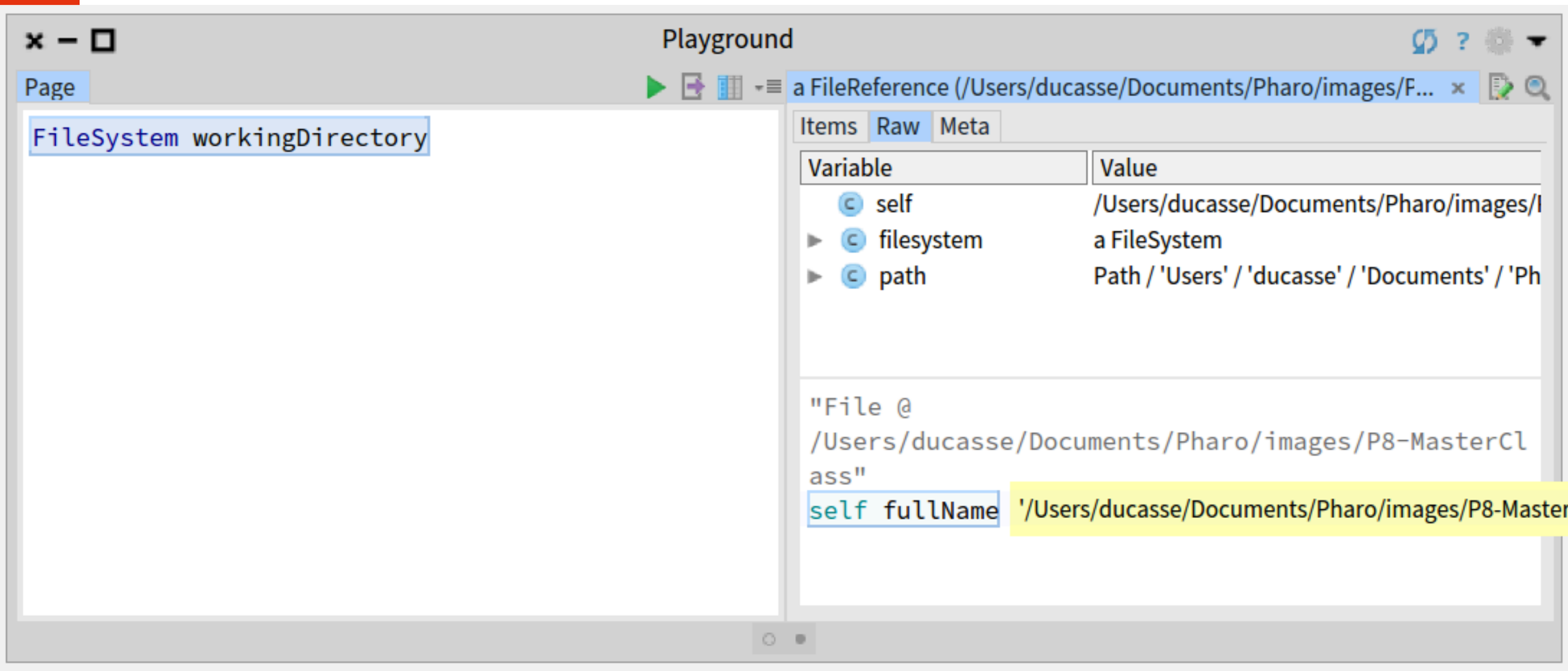
Oh! a file browser in my inspector!



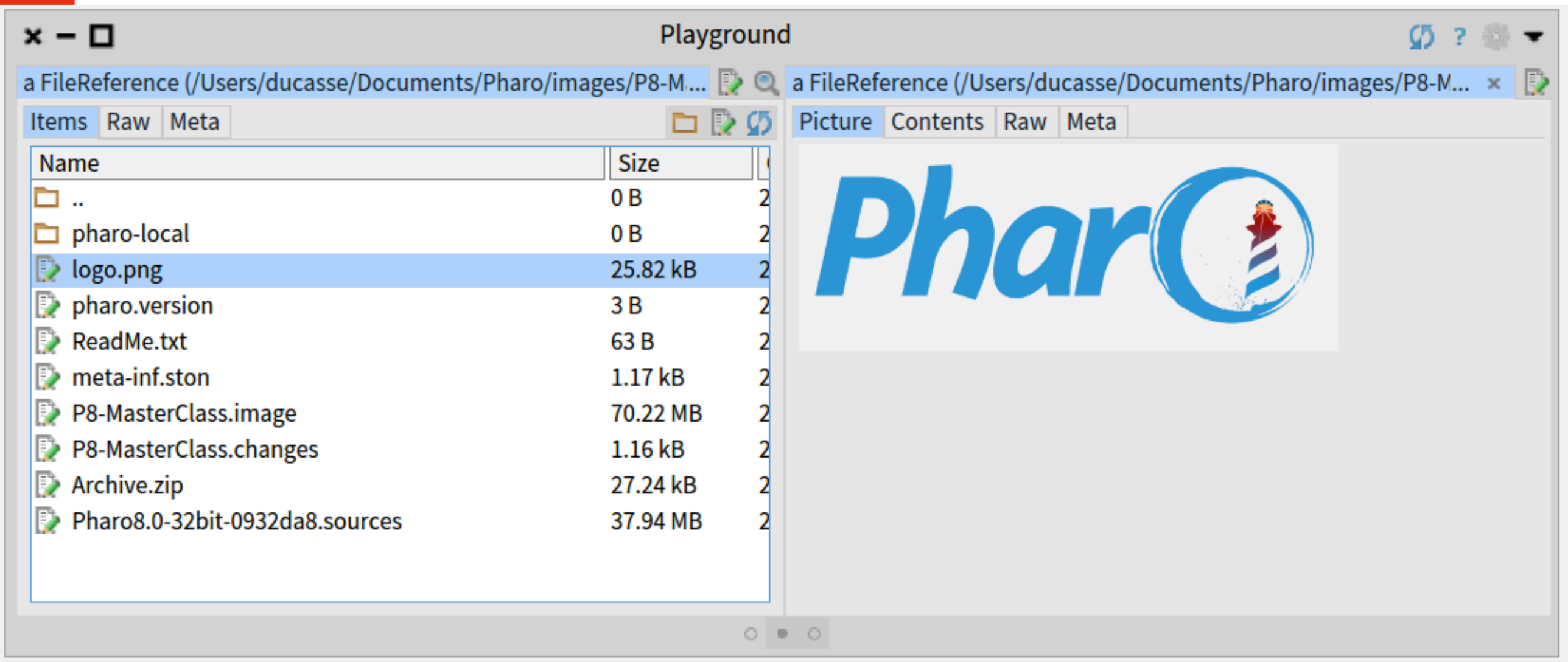
But I have a file reference: a dull object



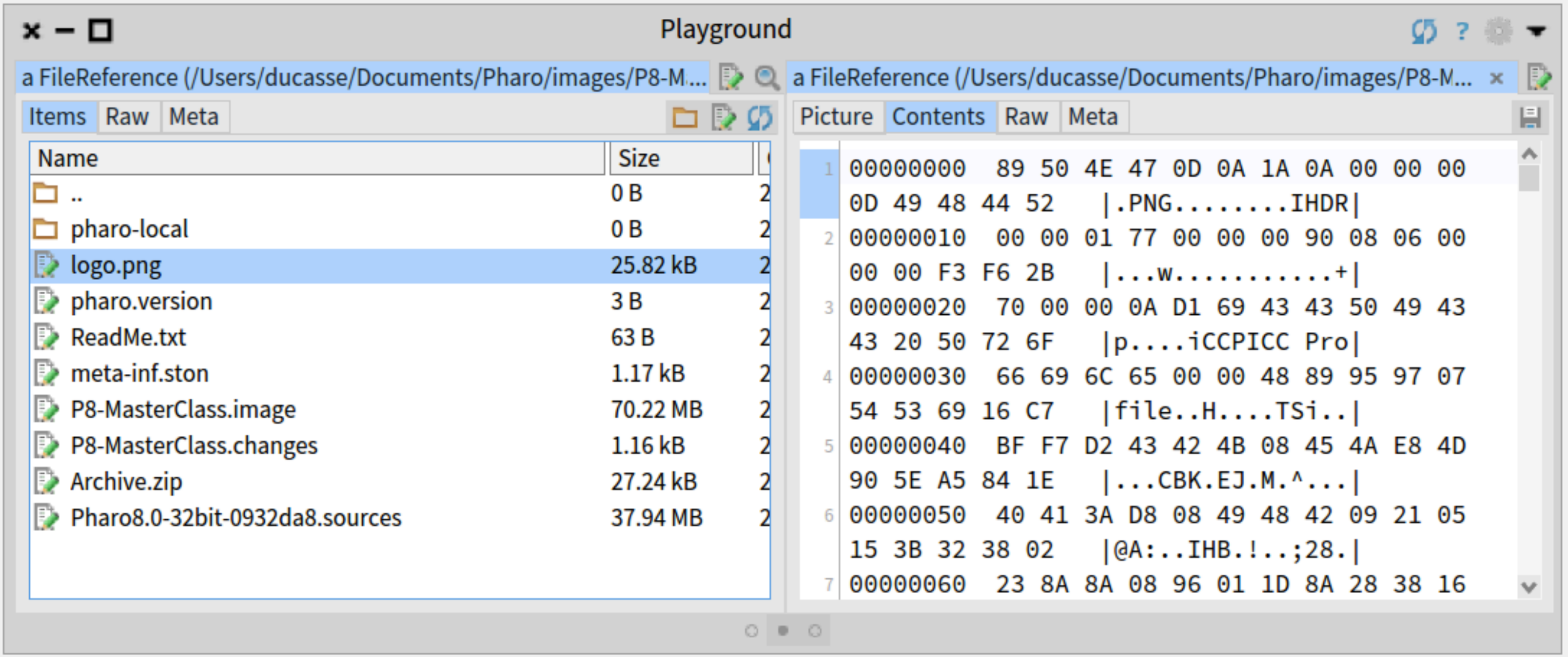
Quite boring object



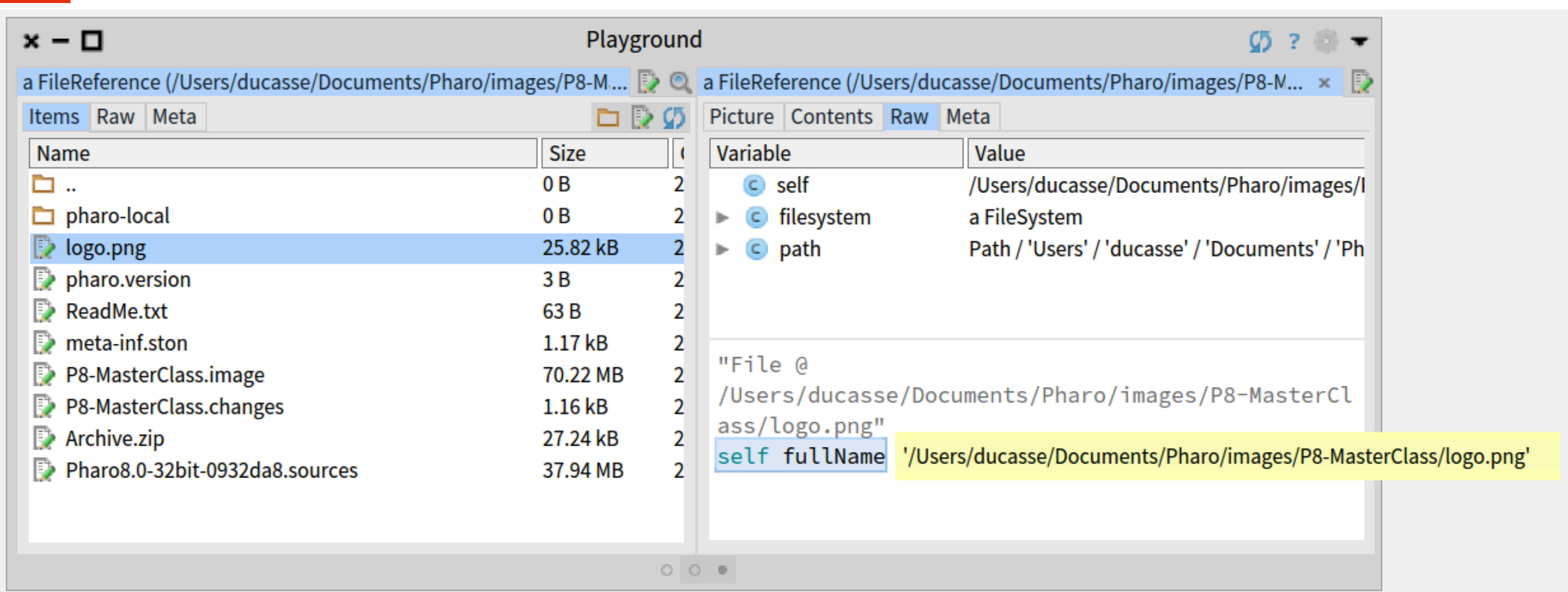
Ah this is this png ;)



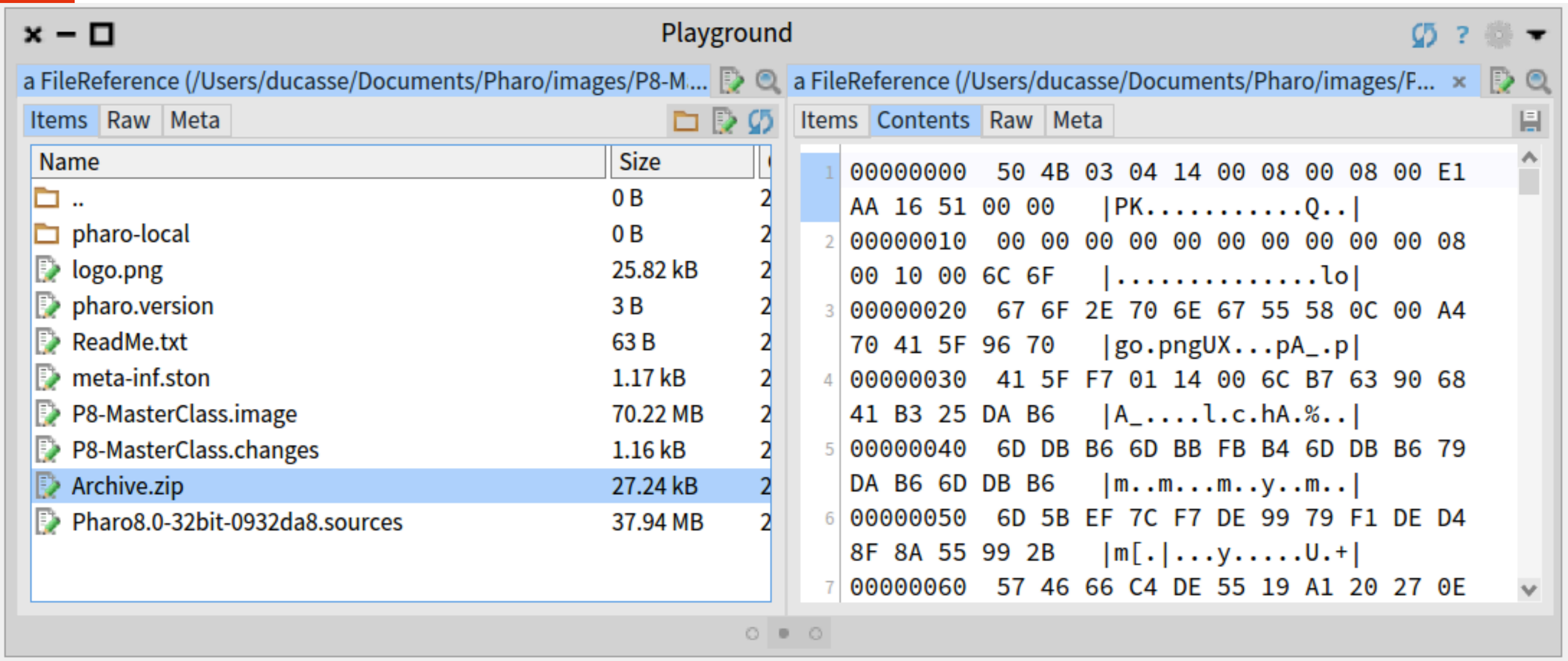
Looking inside that PNG



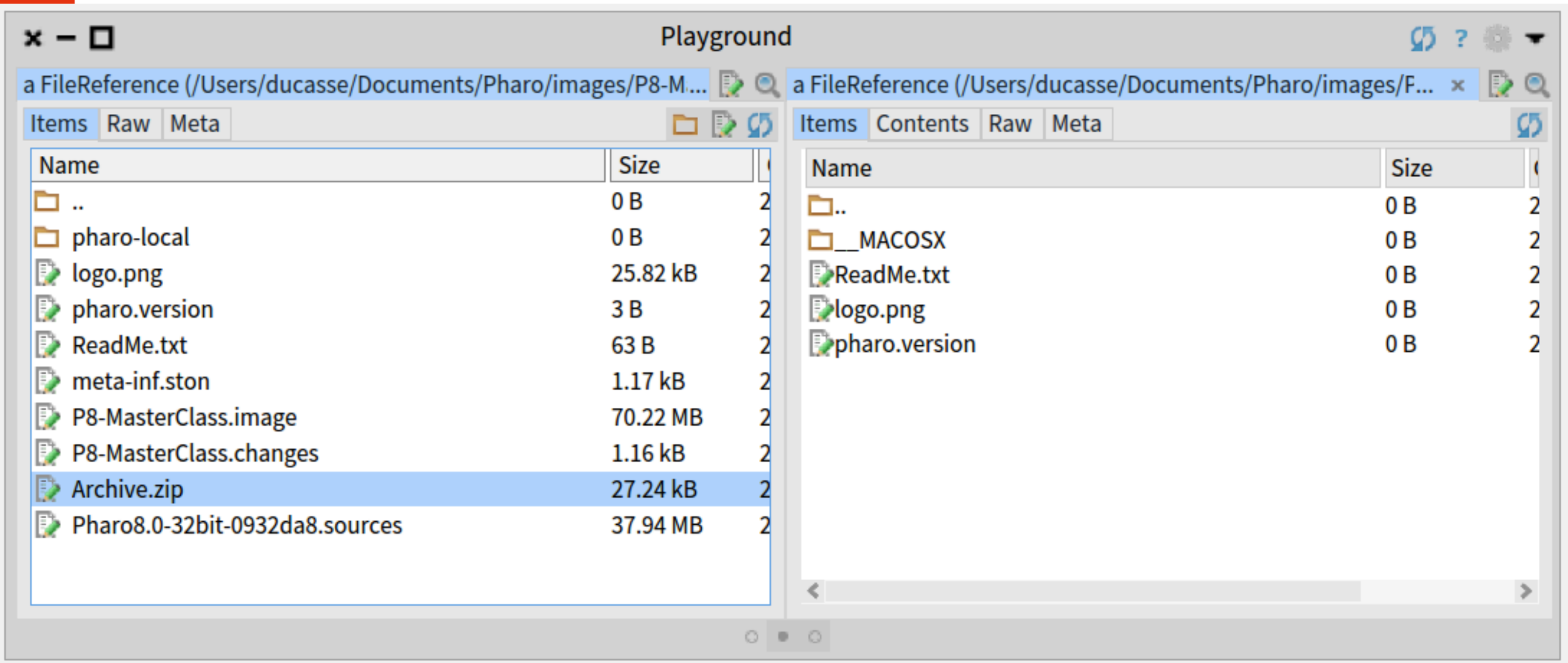
But still a file reference!



See! an archive ‘.zip’



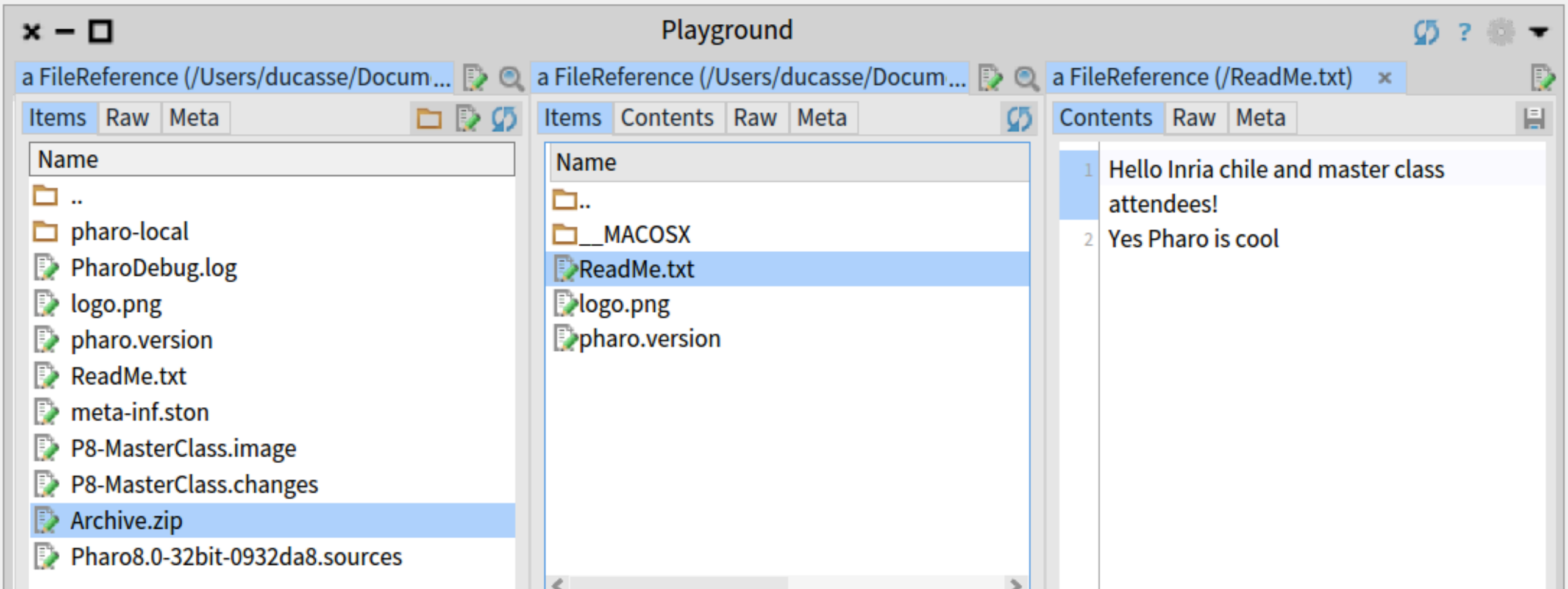
Kind of clear…



An object can expose multiple interactive views!

Users get the best view for your task!

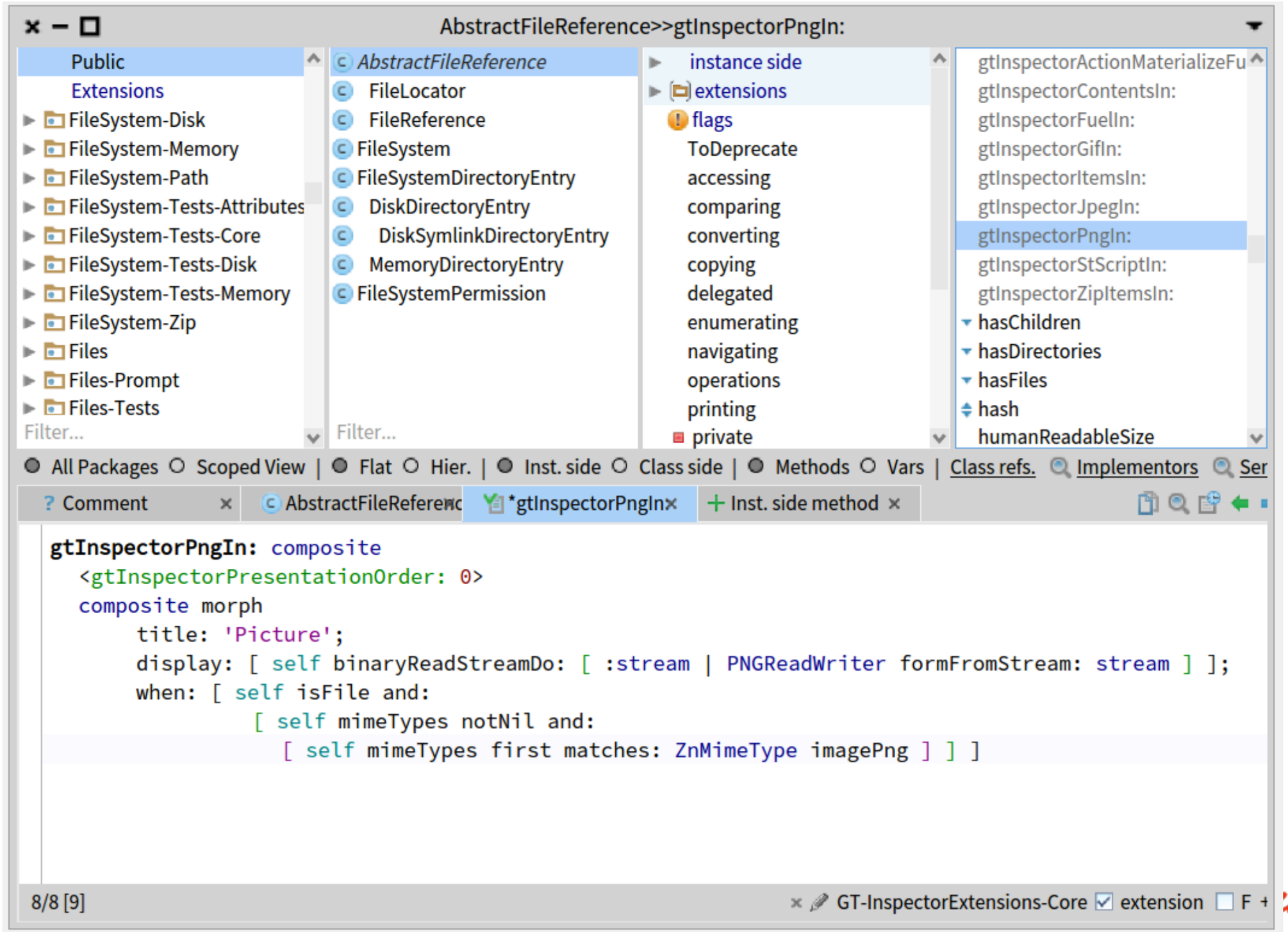
Users can add views to your domain objects



It is cool but it is not magic

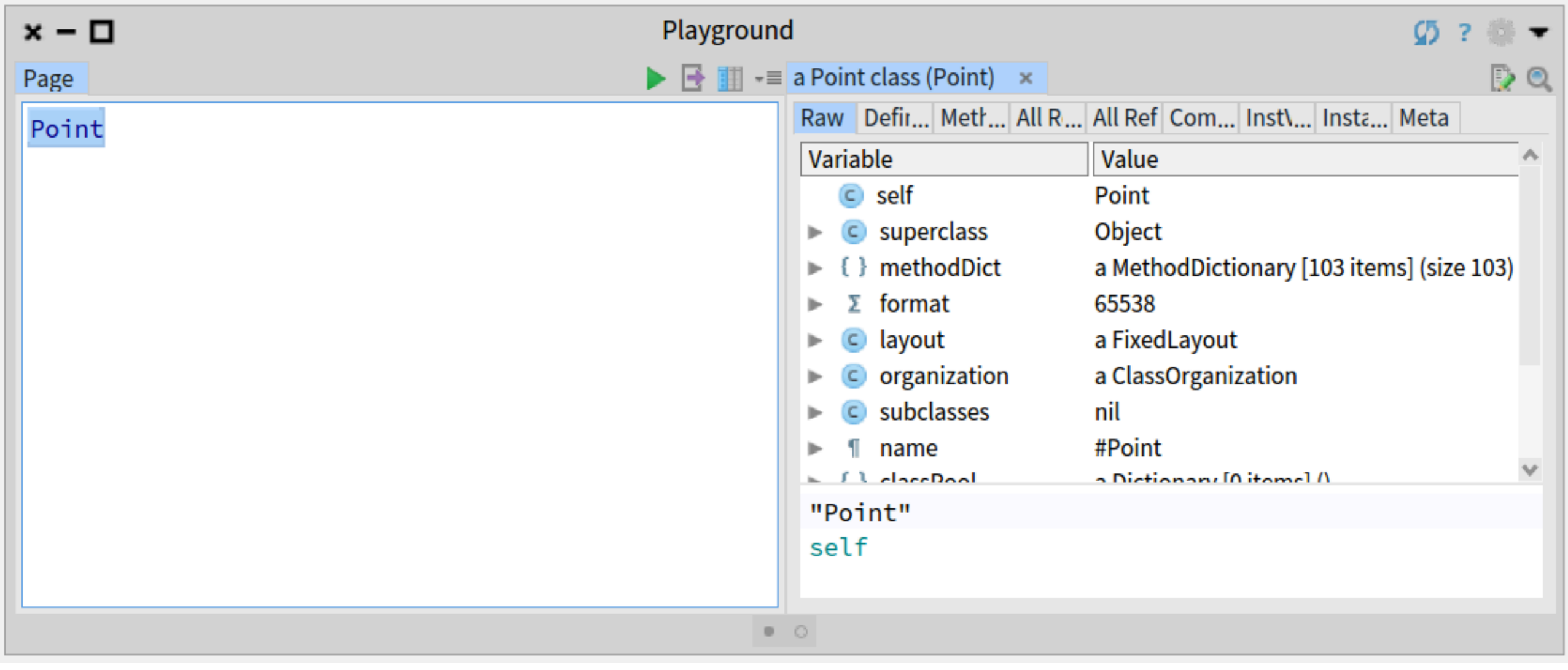
Yes you can define your own!

Implementing a pane!

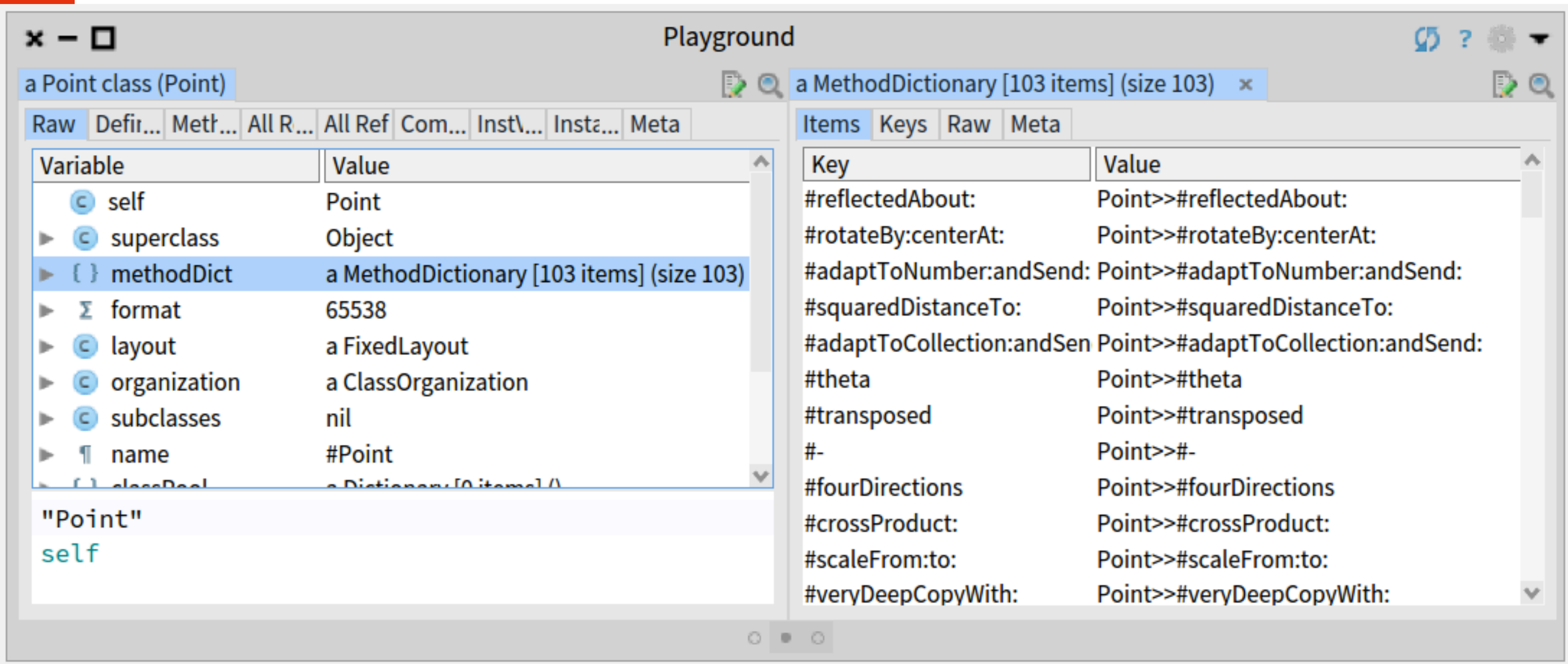


Files are boring… What about *inside* the system?

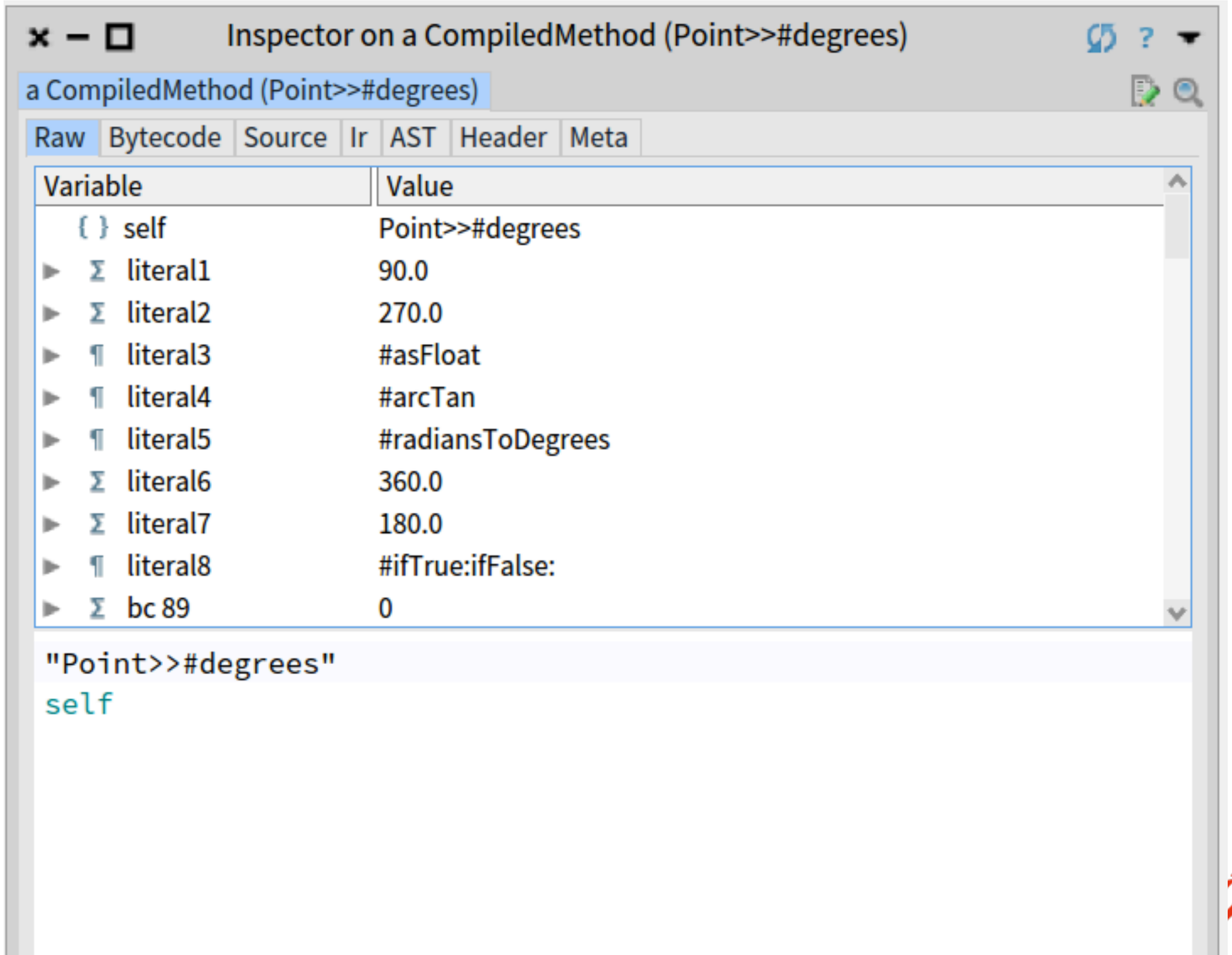
A class is an object we can inspect!



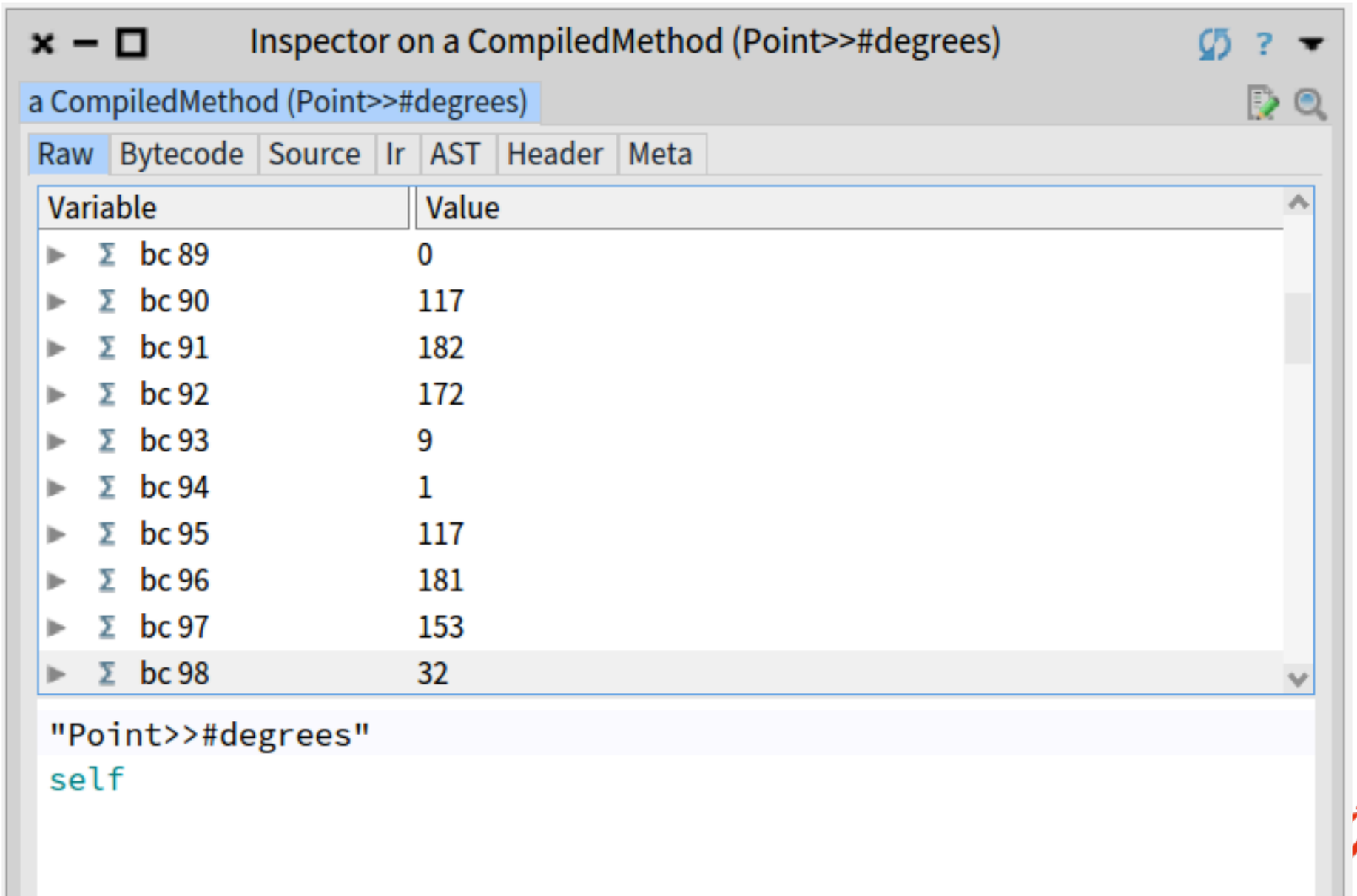
“A class has a method dictionary” they said



Dissecting one method



I do not want to be a decompiler!



It looks like a method

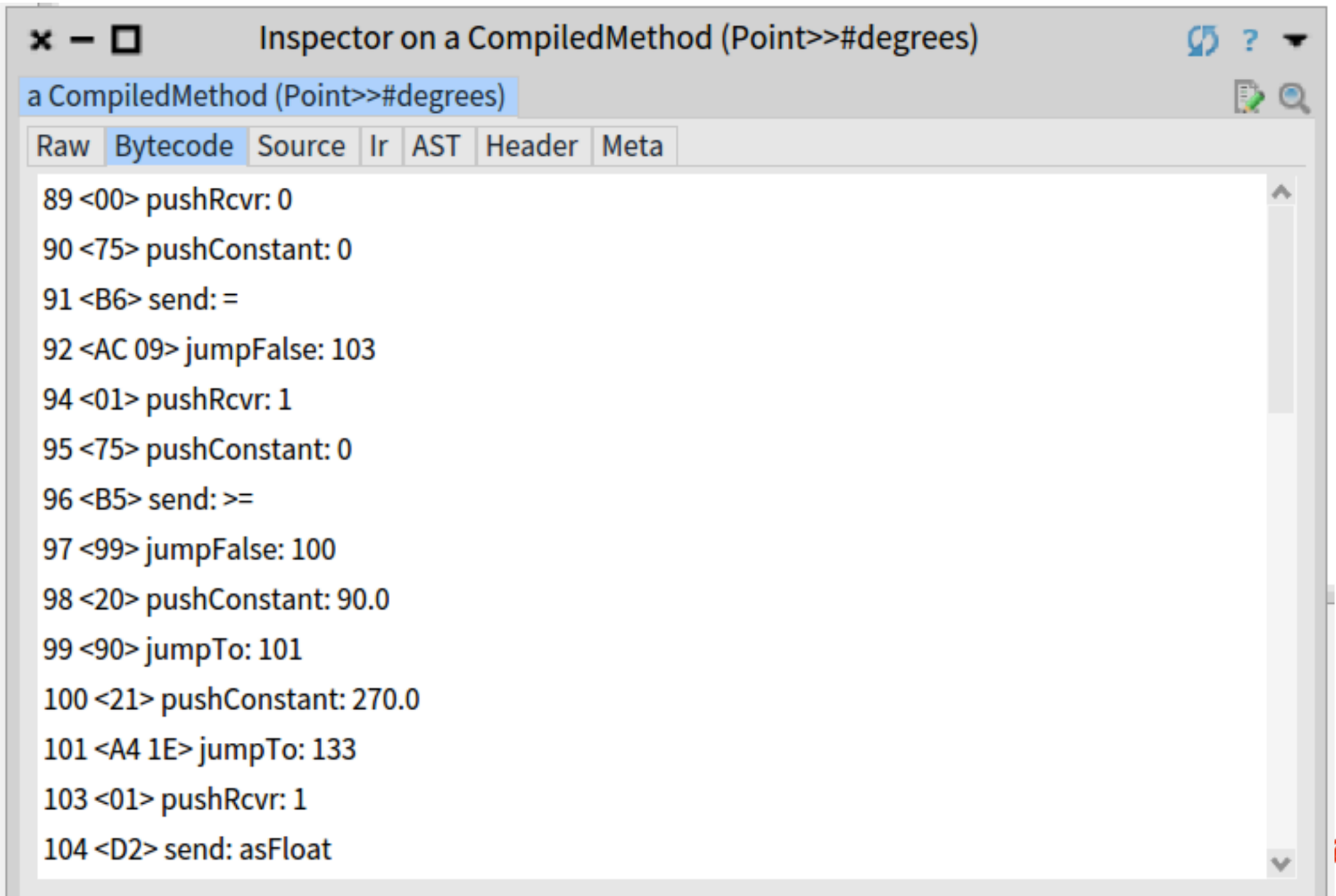
Inspector on a CompiledMethod (Point>>#degrees)

a CompiledMethod (Point>>#degrees)

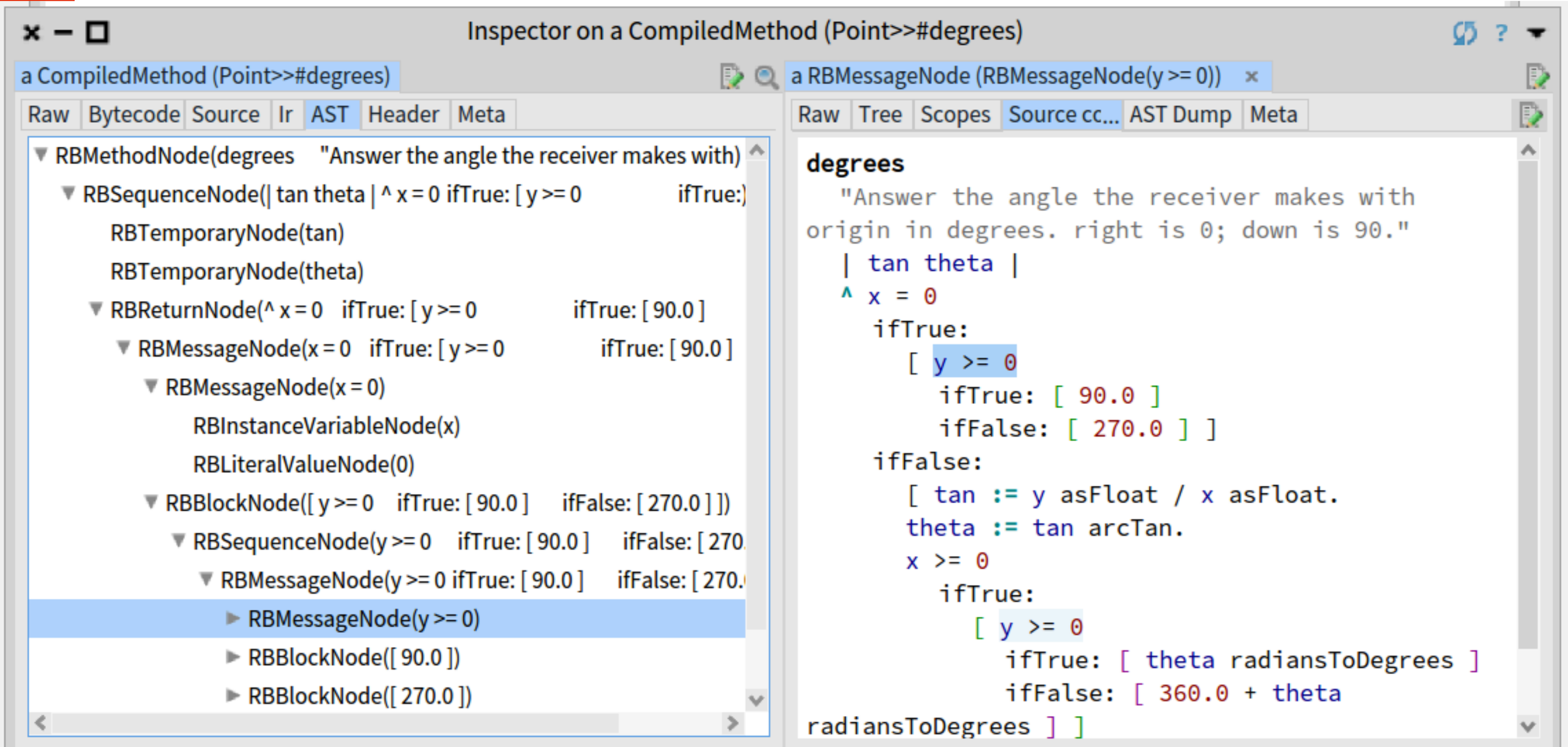
Raw | Bytecode | Source | Ir | AST | Header | Meta

```
degrees
  "Answer the angle the receiver makes with origin in degrees.
right is 0; down is 90."
  | tan theta |
  ^ x = 0
    ifTrue:
      [ y >= 0
        ifTrue: [ 90.0 ]
        ifFalse: [ 270.0 ] ]
    ifFalse:
      [ tan := y asFloat / x asFloat.
      theta := tan arcTan.
      x >= 0
        ifTrue:
          [ y >= 0
            ifTrue: [ theta radiansToDegrees ]
```

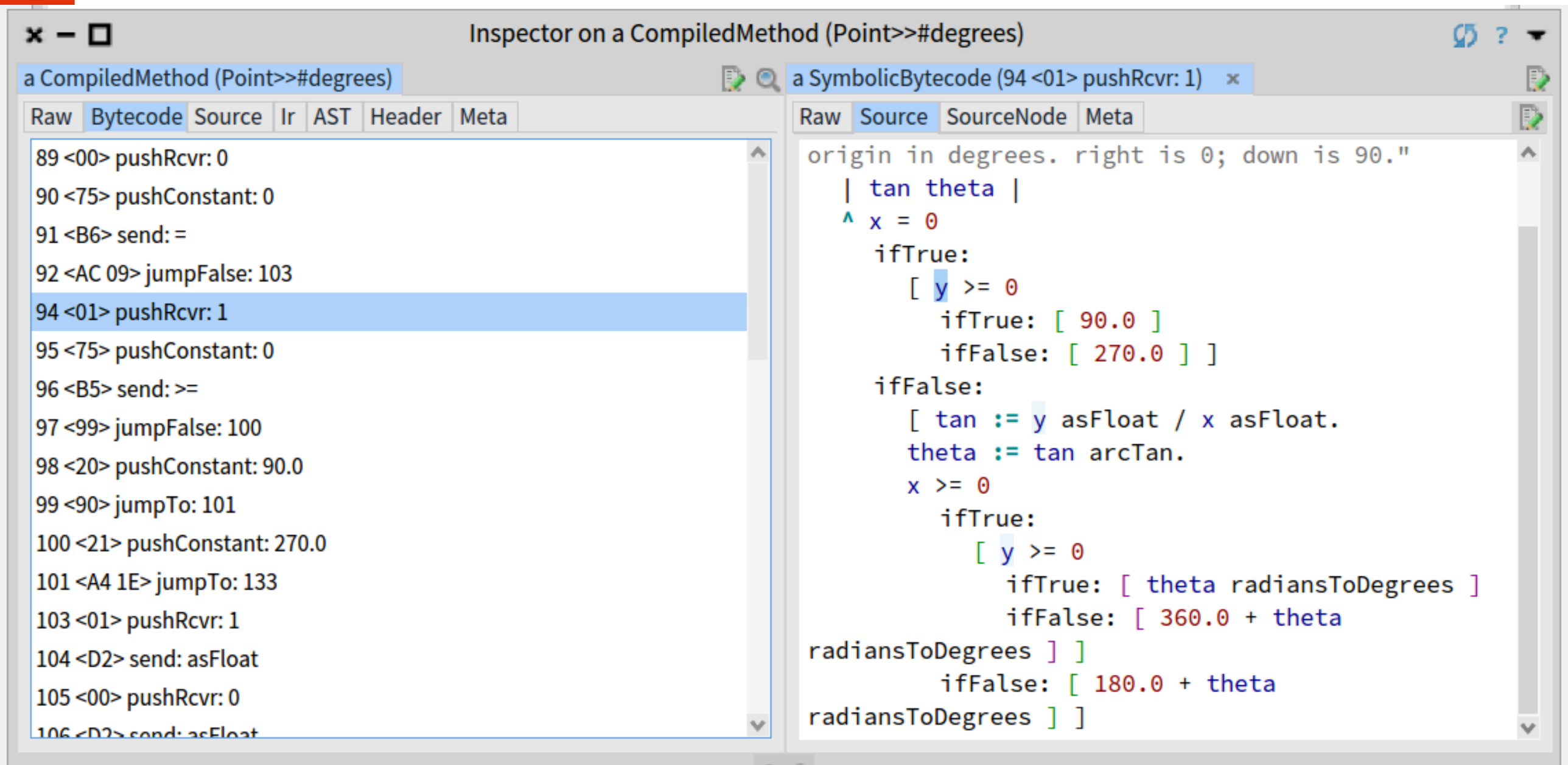
Numbers are not that obscure finally



And mapping them to the right abstraction helps



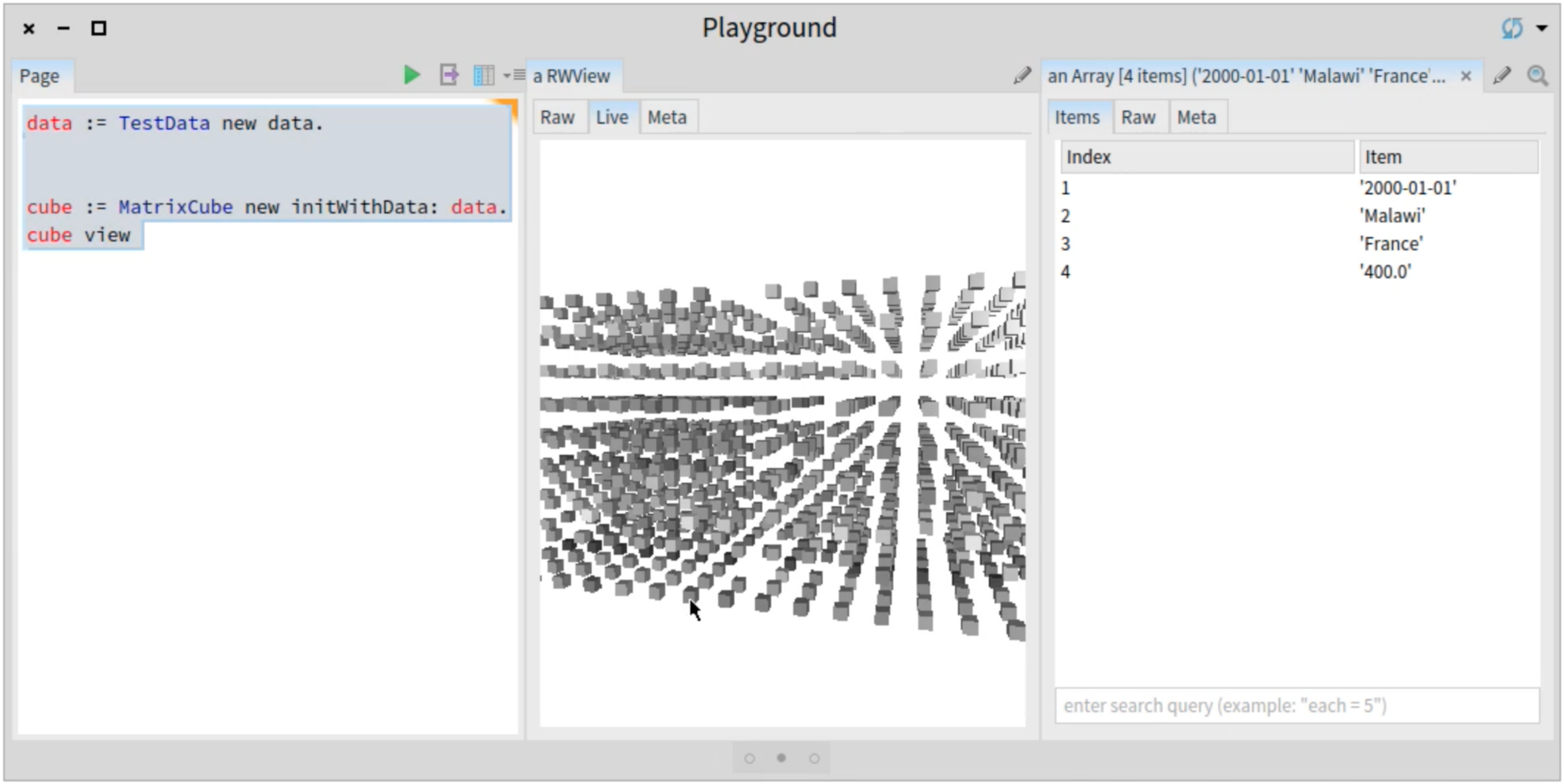
Yes pushRcvr: 1 means the second field!



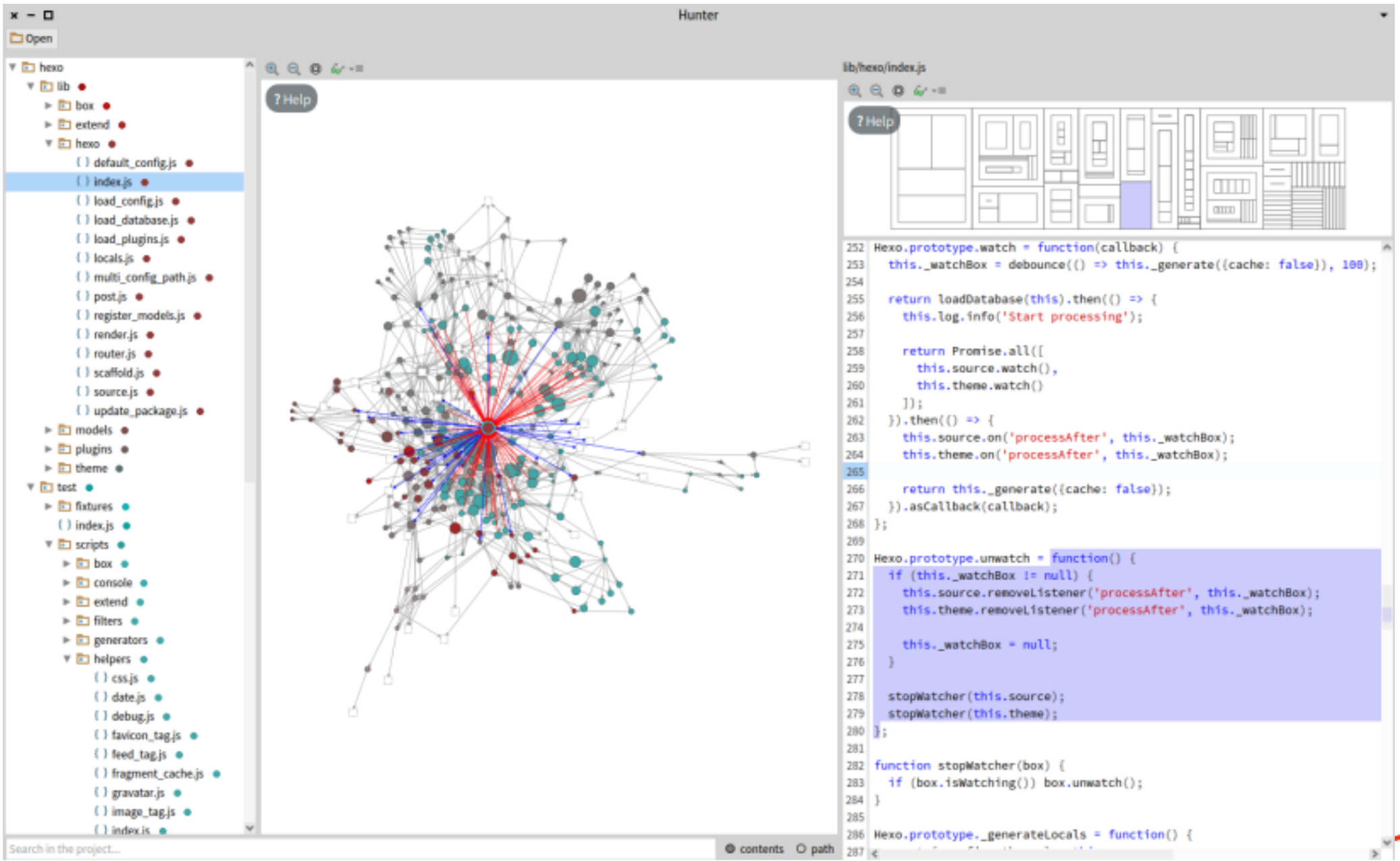
Examples of developer' extensions

Inria

E.g. Inspecting live a 3D object



Javascript analysis



Probabilistic Data Structure

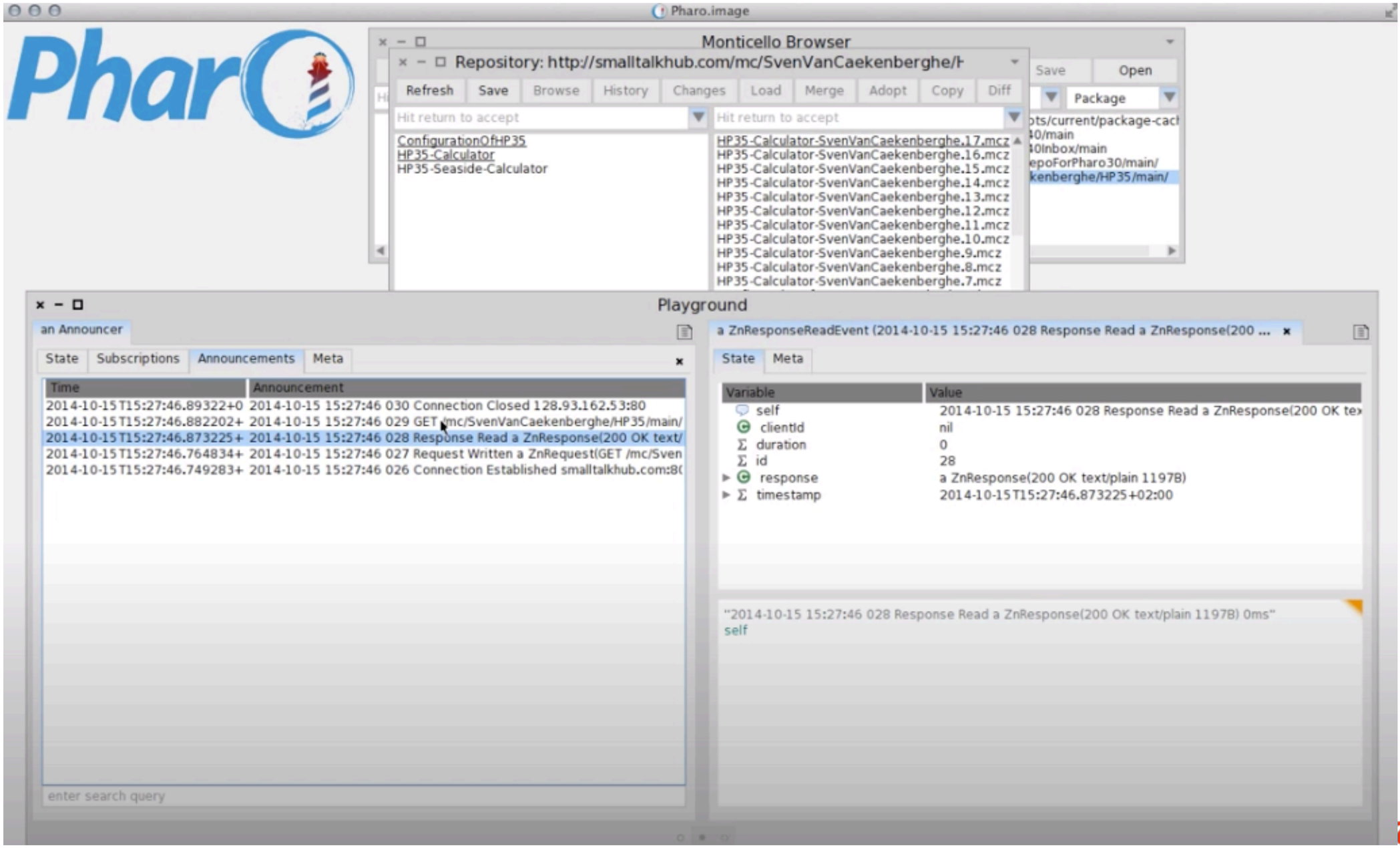
https://github.com/osoco/PharoPDS

Defined new data structure

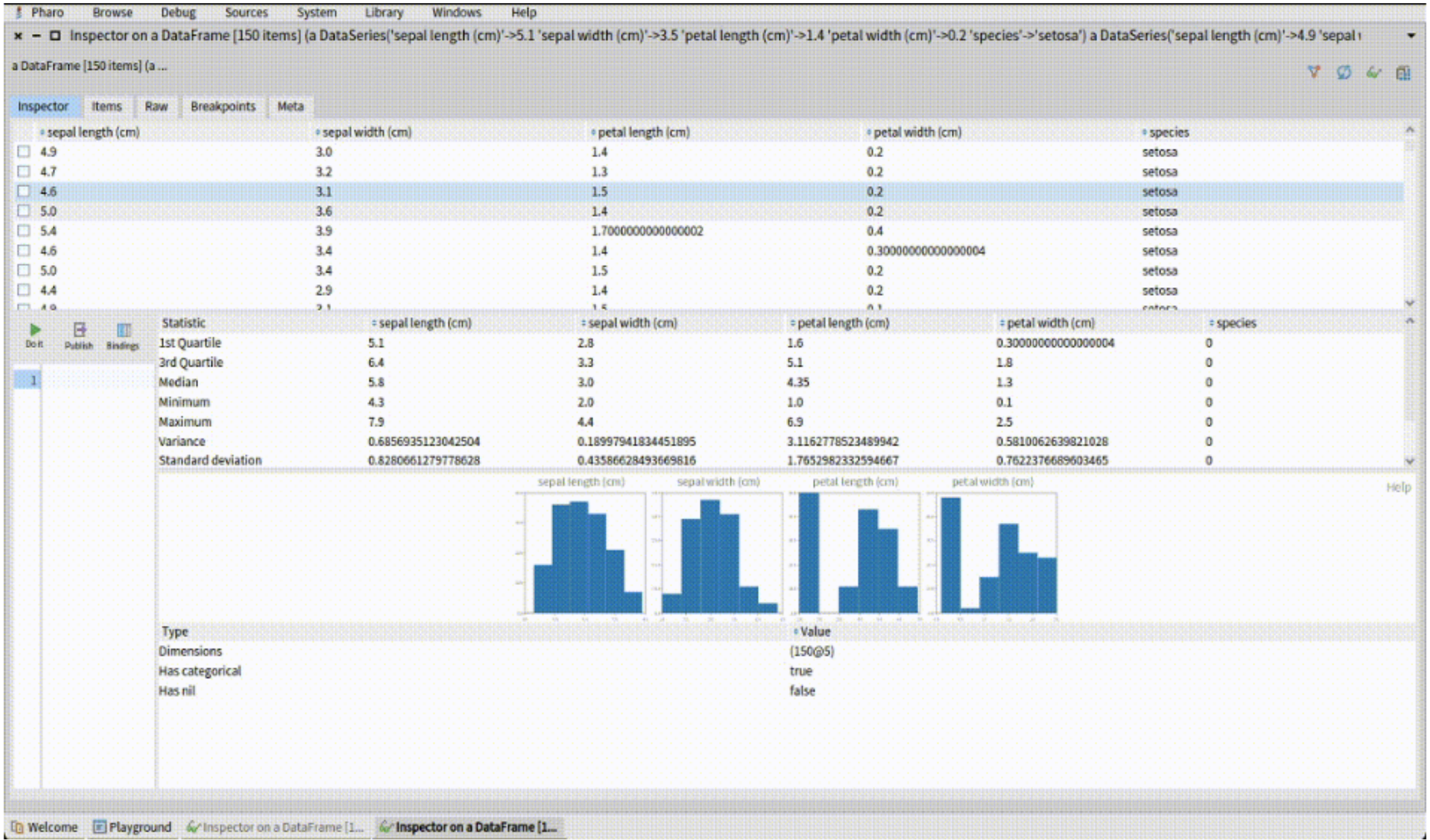
And tools

HTTP traffic analysis

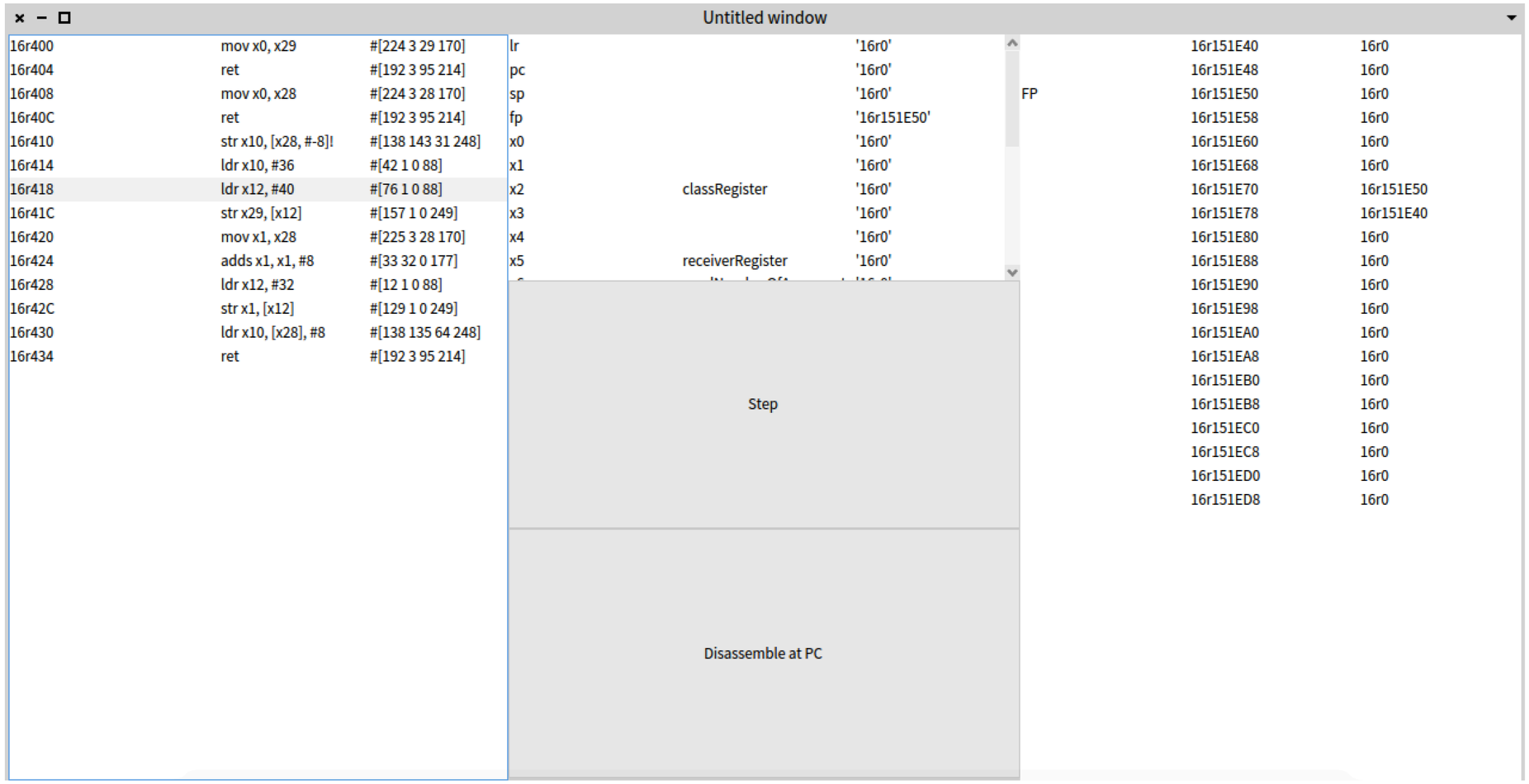
http://youtu.be/rIBbeMdFCys



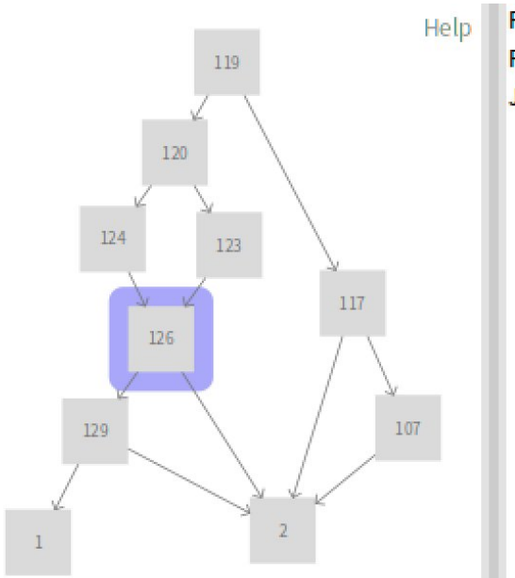
Handling data



Stepping assembly :)



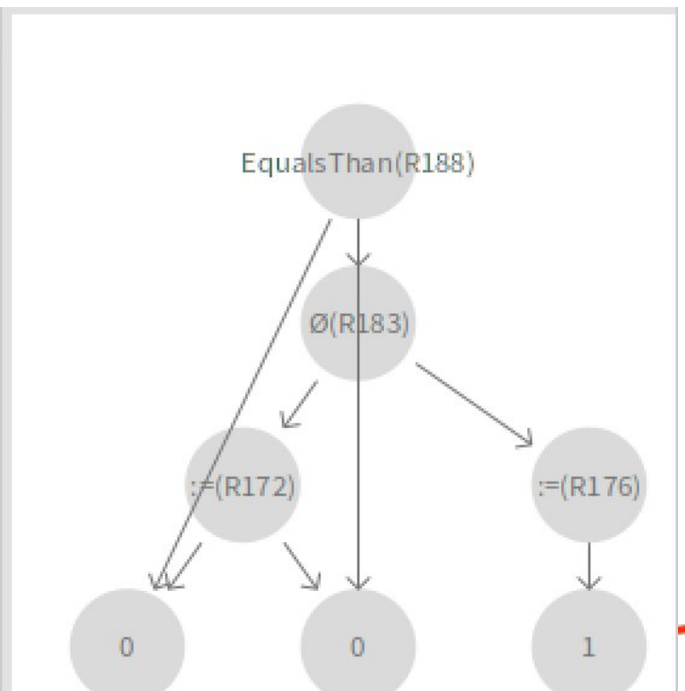
Better than reading Basic Blocks in assembly



Help

R183 := Ø R172 R176
R188 := EqualsThan R183 0
Jump if R188 true -> 2 if false -> 129

R183 := Ø R172 R176
R188 := EqualsThan R183 0
Jump if R188 true -> 2 if false -> 129



Other tools

Hot update on the fly
customizable debugger

QuittanceGenerator class>>#exampleJustMonthAndRenter

Halt

Bytecode

Stack

Proceed Restart Into Over Through

PDFCellElement	getSubElementsWith:styleSheet:
PDFCellElement(PDFComposite)	generateCodeSegmentsCollectionWi
PDFCellElement(PDFComposite)	generateCodeSegmentWith:styleShe
PDFDataTableElement(PDFComposite)	generateCodeSegmentsCollectionWi [:aSubElement | aSubElement generateCodeSe
Array(SequenceableCollection)	collect:

Source

Where is? Browse

```
generateCodeSegmentsCollectionWith: aPDFGenerator styleSheet: compositeStyleSheet format: aFormat
    ^ (self getSubElementsWith: aPDFGenerator styleSheet: compositeStyleSheet)
        collect: [ :aSubElement |
            aSubElement
                generateCodeSegmentWith: aPDFGenerator
                styleSheet: (aSubElement buildCompositeStyleSheetFrom: compositeStyleSheet)
                format: aFormat ]
```

Variables

Type	Variable	Value
implicit	self	a PDFCellElement
parameter	aFormat	a PDFA4Format
parameter	aPDFGenerator	a PDFGenerator
parameter	compositeStyleSheet	a StyleSheet

```
        dimension: 80 mm @ 20 mm;
```

debug it

Bytecode GT

Parser Stack | Through Action | Into Reduce | Next Token | Possible Actions | GLR

Symbol	Value
Expression	a JSUnaryExpressionNode
")"	{)(238498,238498,#(23))}
"{"	{{(238499,238499,#(25))}
StatementList	an OrderedCollection(a JSExpressionStatementNode a JSI
"("	{((238678,238678,#(28))}
"function"	{function(238679,238686,#(53 82))}
Identifieropt	nil
"("	{((238687,238687,#(28))}
FormalParameterListopt	an Array(an OrderedCollection() an OrderedCollection())
")"	{)(238688,238688,#(23))}
"{"	{{(238689,238689,#(25))}
SourceElements	an OrderedCollection(a JSVariableStatementNode a JSVar
MemberExpression	a JSNameExpressionNode
"."	{.(274447,274447,#(38))}

Symbol	Action
<identifier>	Shi
IdentifierName	Shi

1

A specific parser compiler debugger

Input | Step to Cur | Next Character

```
            return t._stash(this); // dojo.NodeList
        }

    });

    nl.events = [
        // summary:
        //          list of all DOM events used in NodeList
        "blur", "focus", "change", "click", "error", "keydown", "keypress",
        "keyup", "load", "mousedown", "mouseenter", "mouseleave", "mousemove",
        "mouseout", "mouseover", "mouseup", "submit"
    ];

    // FIXME: pseudo-doc the above automatically generated on-event functions

    // syntactic sugar for DOM events
    d.forEach(nl.events, function(evt){
```

Nam	Value
State	default

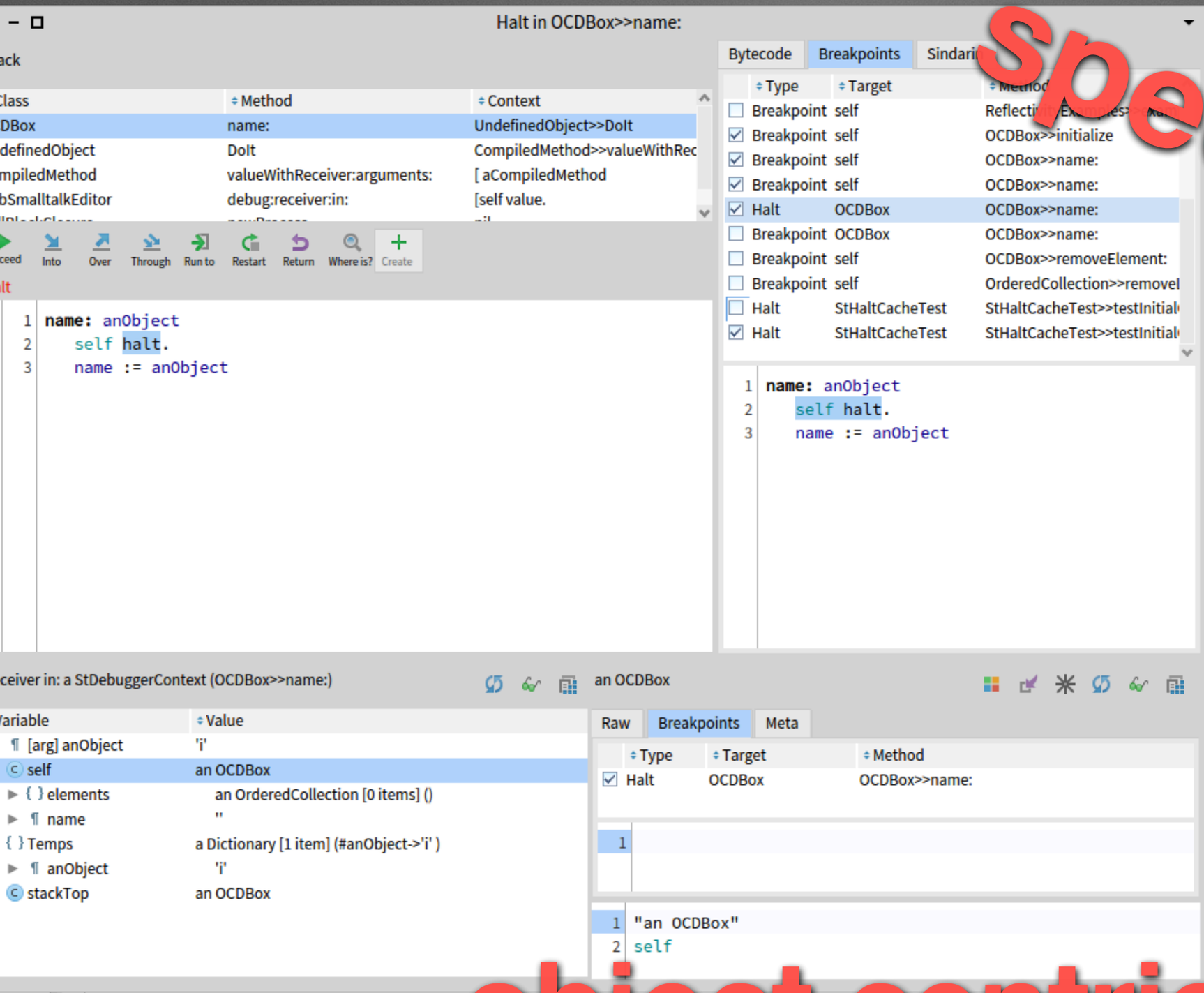
Stack | Proceed | Restart | Into | Over | Through | Source | Where is? | Browse

JSParser(SmaCCGLRParser)	performParsingStep	
JSParser(SmaCCGLRParser)	performParsingLoop	
JSParser(SmaCCParser)	parse	
JSParser class(SmaCCParser class)	parseStream:startingAt:	
JSParser class(SmaCCParser class)	parseFile:	[self parseStrean
BlockClosure	ensure:	
JSParser class(SmaCCParser class)	parseFile:	
UndefinedObject	DoIt	
CompiledMethod	valueWithReceiver:arguments:	
RubSmalltalkEditor	debug:receiver:in:	[aCompiledMetho
BlockClosure	newProcess	[self value.Proces

```
(tryAllStates and: [ self isEOFToken ])
    ifTrue: [ ^ self processEOFToken ].
currentToken isPattern
    ifTrue: [ ^ self splitForPatternToken ].
[ actions := self actionsForCurrentToken.
actions isEmpty
    ifTrue: [ self killState.
        currentToken := nil.
        ^ self ].
2 to: actions size do: [ :i | self splitWithAction: (actions at: i) ].
action := actions first.
action = self acceptAction
    or: [ self performAction: action.
        currentToken isNil ] ] whileFalse.
action = self acceptAction
    ifTrue: [ currentState markAccepted ]
```

Specific view

object centric + adv scripting (AST + call stack)



A little moment of grace…

Requirements

I want my breakpoint to only stop

when called from THAT test called

testMe!

Basically a control flow breakpoint

Inria

E.g.

```
mycode

    self haltIf: #testMe
    ...
```

Implementation

Use stack reification

Walk it

Halt if needed

(in 5 lines)

```
haltIf: aSelector
    | cntxt |
    cntxt := thisContext.
    [ cntxt isNil ] whileFalse: [
        cntxt selector = aSelector
          ifTrue: [ self halt ].
        cntxt := cntxt sender ]
```

Inria

Pharo

Simple and powerful

Productive

Adaptable