b
UNIVERSITÄT
BERN

Adaptable Visualisation based on user needs

Leonel Merino
merino @ iam.unibe.ch

Software Composition Group
University of Bern

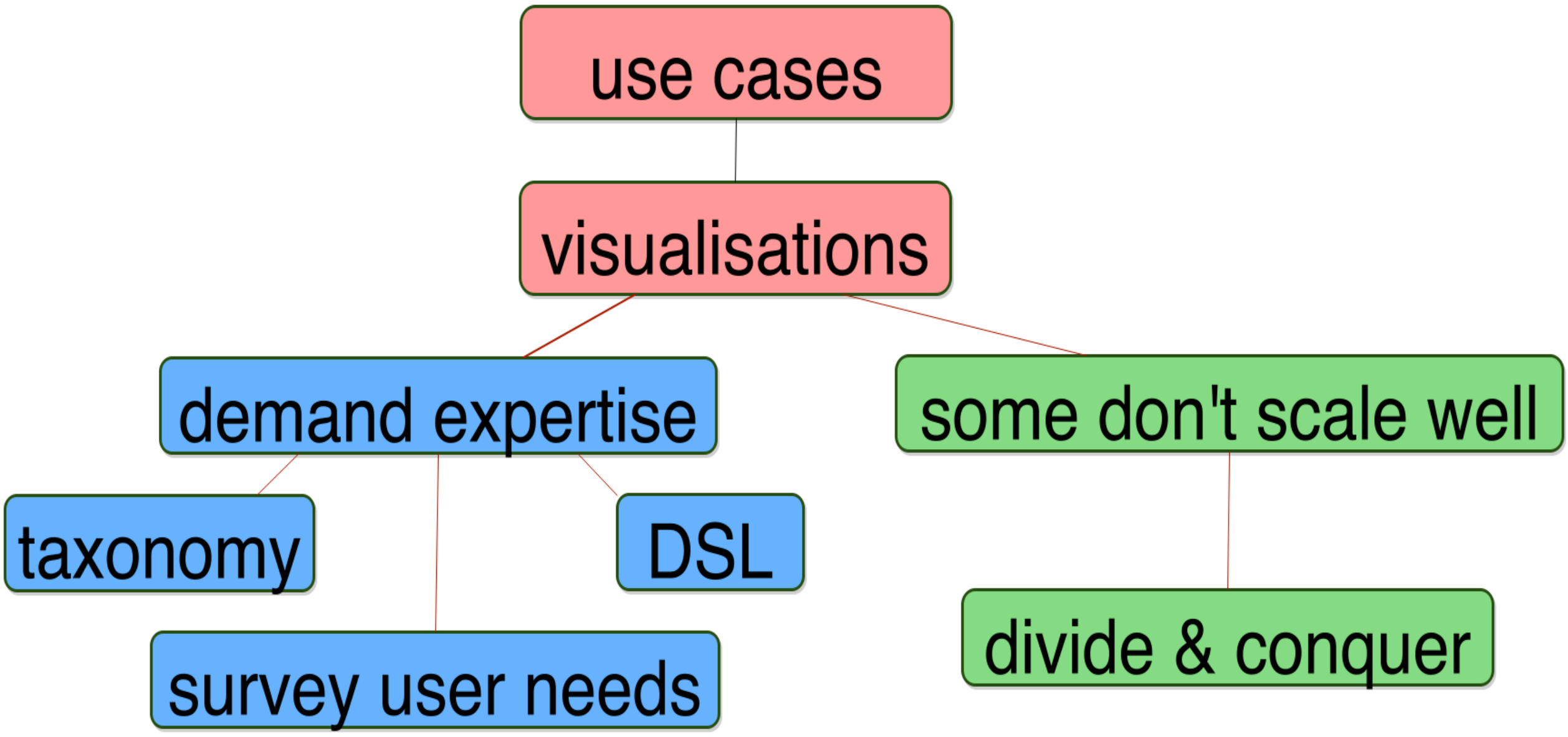
Leonel Merino

Chilean

Who am I?

- Bachelor from University of Chile
  - Working at industry
- Master of Science (EMN - VUB)
  - Working at industry
- March 2014 PhD student at SCG - U. of Bern

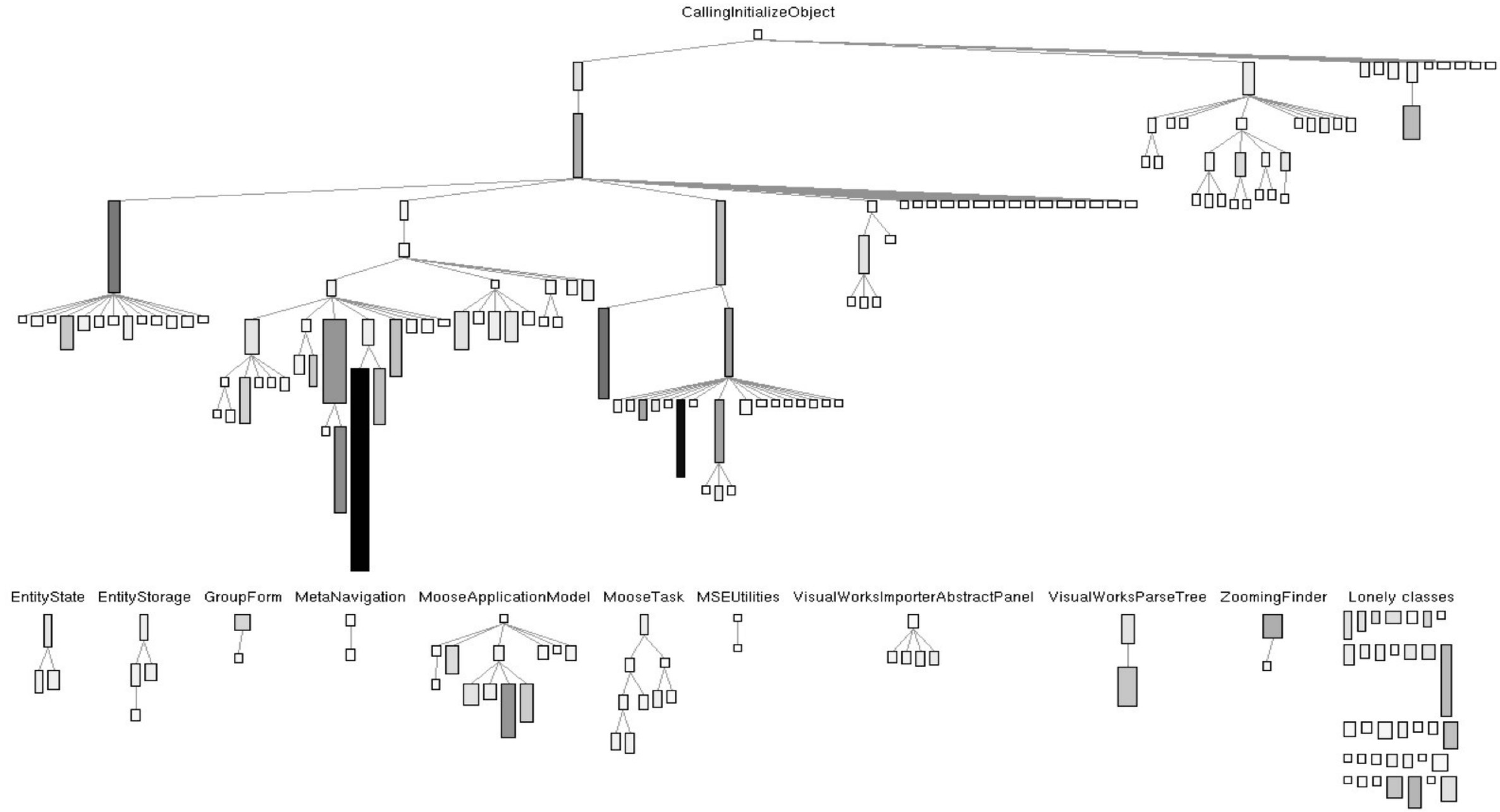
Agenda


use cases
visualisations
demand expertise
some don't scale well
taxonomy
DSL
survey user needs
divide & conquer

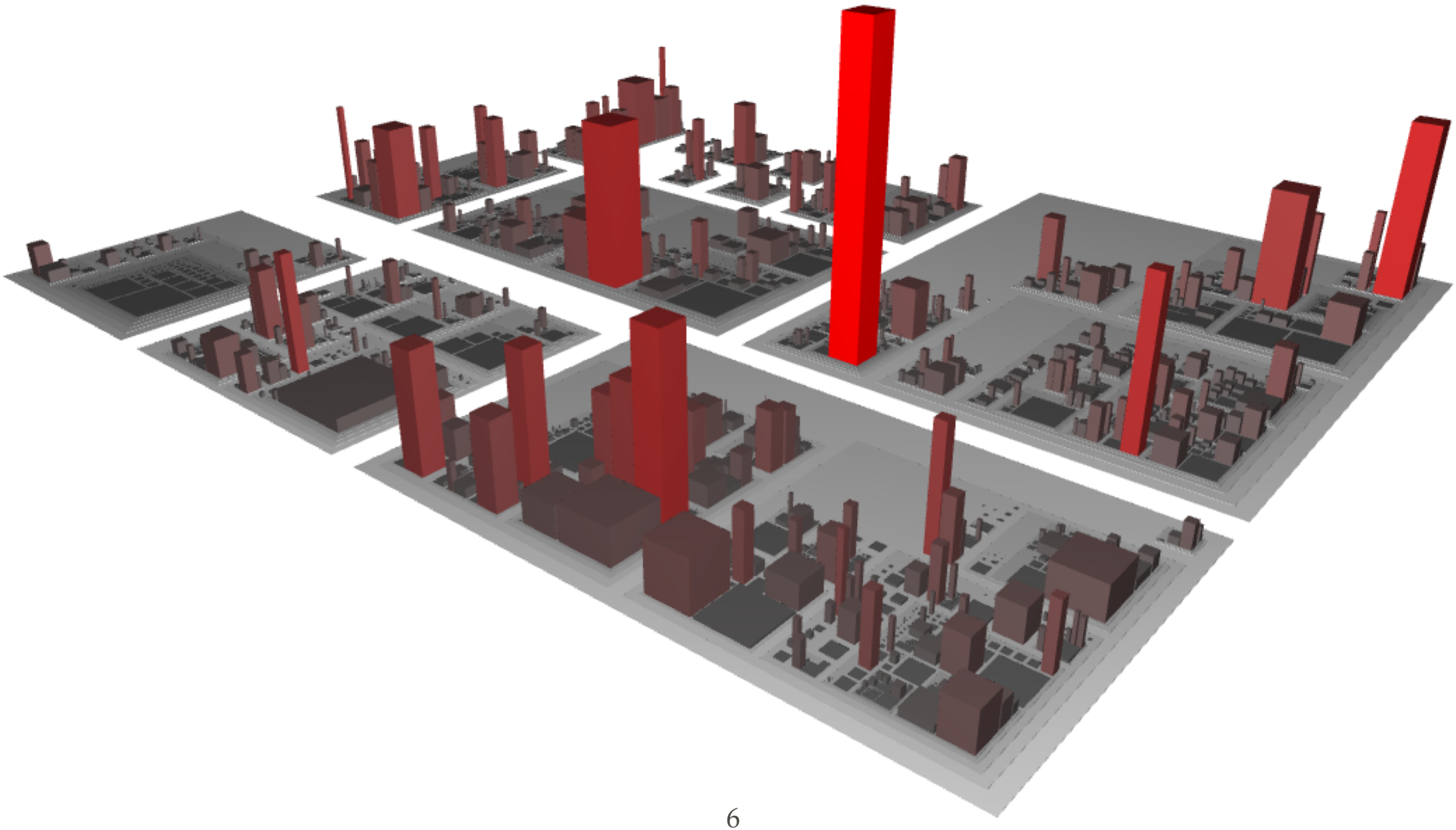
Users need to see code in different ways

Use case: a researcher wants to analyse polymorphism and call sites at class level on a corpus.

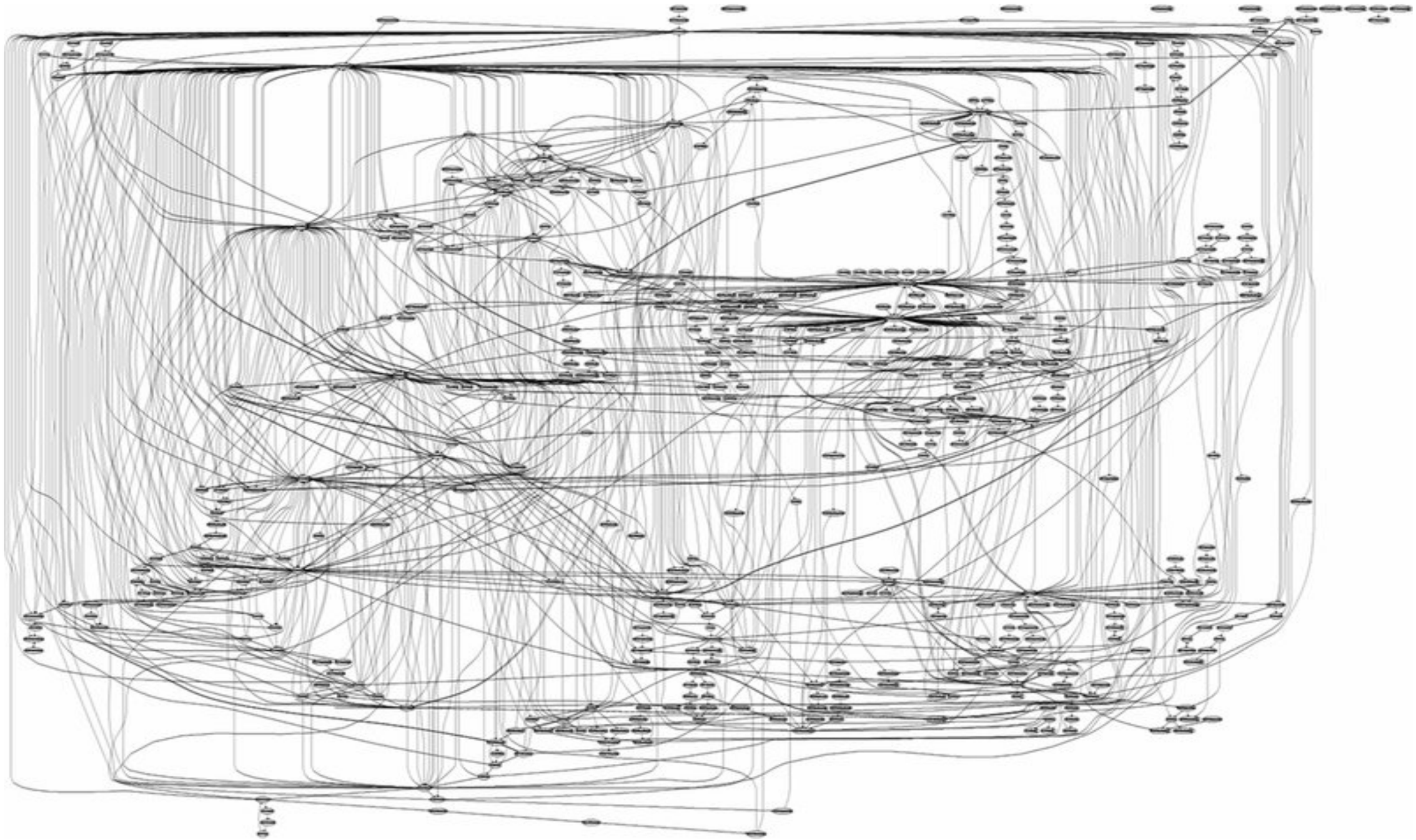
Visualisations give context



Visualisations may simplify complexity

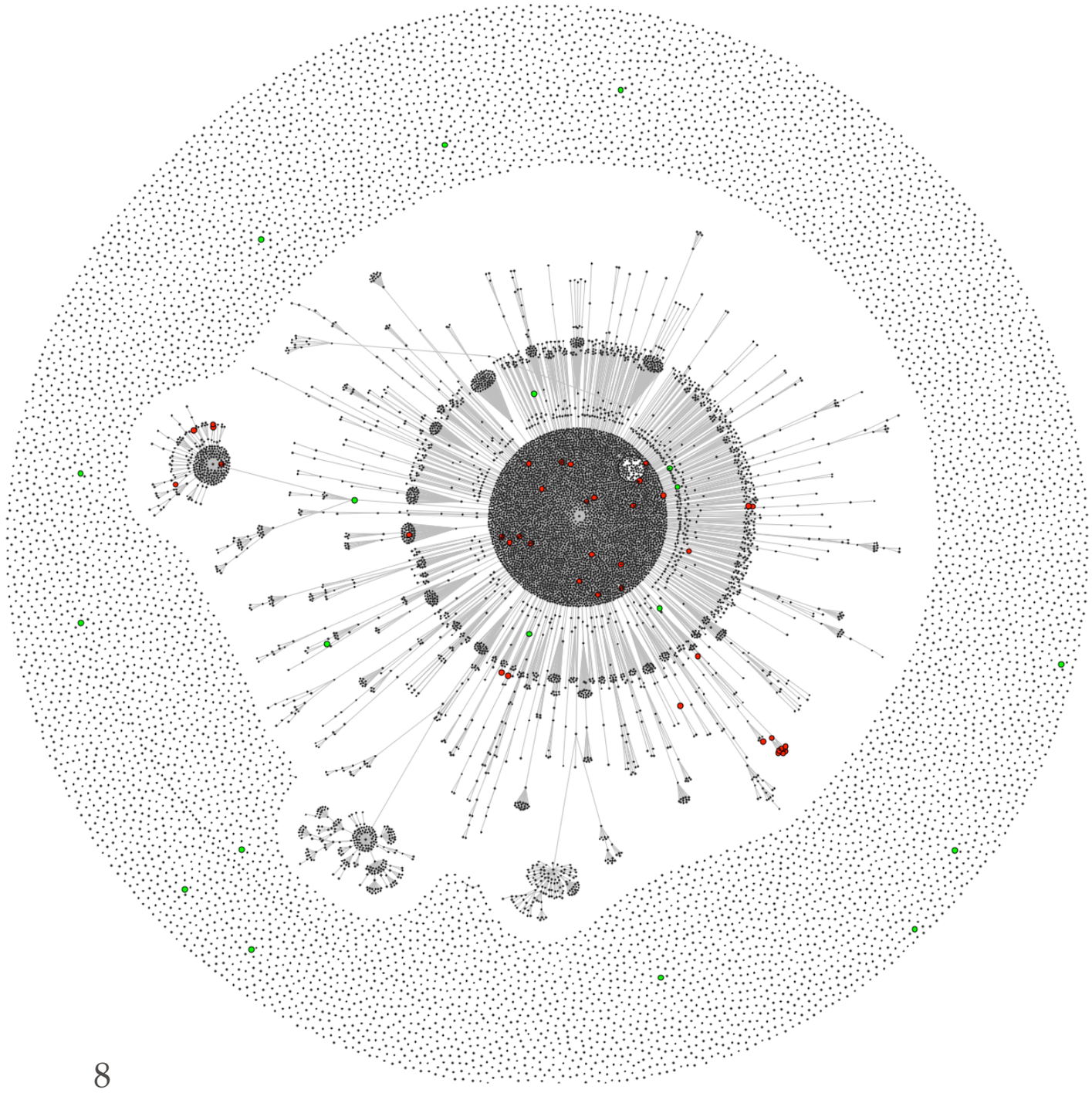


Visualisations not always make things simple



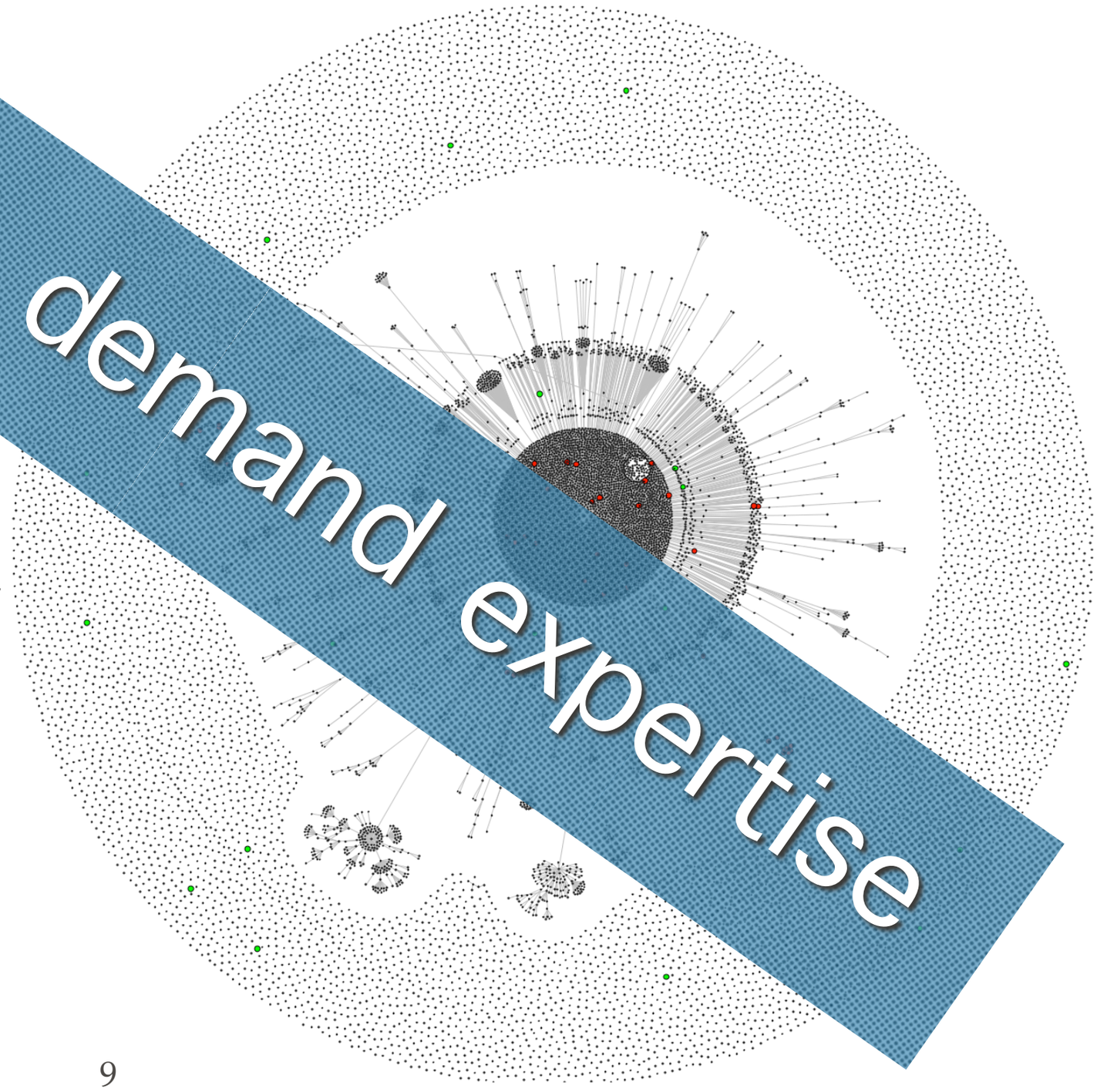
Another way to do the same

```
| aROView view windowSize|
windowSize :=2400@1300.
aROView := ROView new.
view := ROMondrianViewBuilder view: aROView.
aView shape circle
if: [ :aClass | self isPolyClass: aClass ] fillColor: Color green;
if: [ :aClass | self isCallingPolyClass: aClass ] fillColor: Color red;
size: [ :aClass | self getSize: aClass ].
aView interaction
on: ROMouseClick
do: [ :event |
	| m browser |
	m := event element model.
	(self isPolyClass: m)
	ifTrue: [ self viewMethodBuilderFocusedOnAPolyClass: m ]
	ifFalse: [
		(self isCallingPolyClass: m)
		ifTrue: [ self viewMethodBuilderFocusedOnACallSite: m ]
		ifFalse: [
		browser := MooseNamespacesCodeBrowser new browser.
		browser openOn: (m mooseModel allNamespaces select: #isRoot).
		(browser pane port: #focusOnClass) value: m ] ] ];
popupText: [ :c |
	| poly |
	poly := self buildPopupText: c.
	c name , ' (' , (self getSize: c) asInteger printString , 'x)'
	,
	(poly size > 0
	ifTrue: [ String cr , '---calls---' , String cr , (String cr join: poly) ]
	ifFalse: [ '' ]) ].
aView nodes: model allClasses.
aView edges: model allClasses from: [ :node | node ] to: [ :node | node
superclass ].
aView forceBasedLayout
ROEaselMorphic new populateMenuOn: view.
view openInWindowSized: windowSize.
```



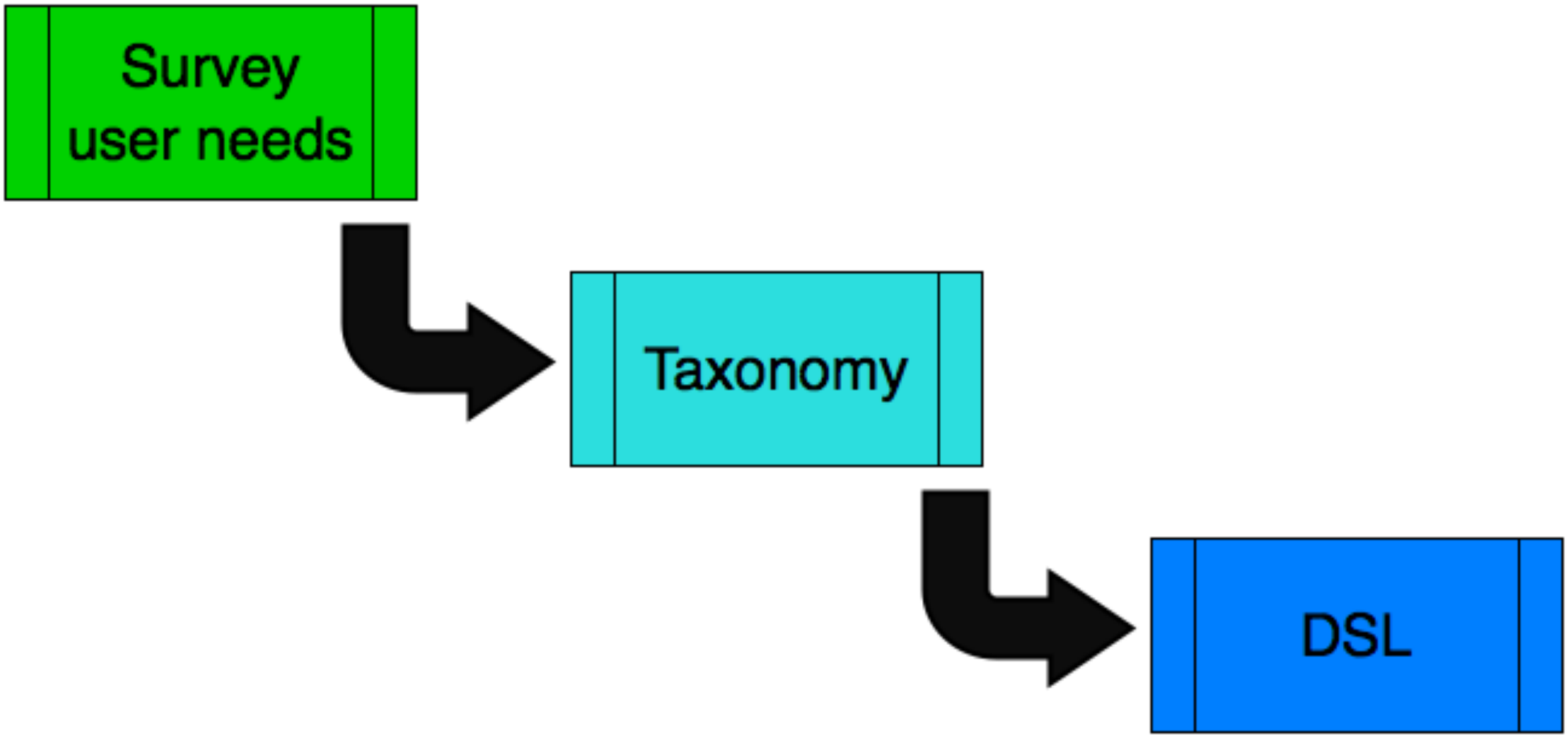
Another way to do the same

```
| aROView view windowSize|
windowSize :=2400@1300.
aROView := ROView new.
view := ROMondrianViewBuilder view: aROView.
aView shape circle
if: [ :aClass | self isPolyClass: aClass ] fillColor: Color green.
if: [ :aClass | self isCallingPolyClass: aClass ] fillColor: Color red.
size: [ :aClass | self getSize: aClass ].
aView interaction
on: ROMouseClick
do: [ :event |
        | m browser |
        m := event element model.
        (self isPolyClass: m)
        ifTrue: [ self viewMethodBuilderFocusedOnAPolyClass: m ]
        ifFalse: [
                (self isCallingPolyClass: m)
                ifTrue: [ self viewMethodBuilderFocusedOnACallSite: m ]
                ifFalse: [
                browser := MooseNamespacesCodeBrowser new browser.
                browser openOn: (m mooseModel allNamespaces select: #isRoot).
                (browser pane port: #focusOnClass) value: m ] ] ];
popupText: [ :c |
        | poly |
        poly := self buildPopupText: c.
        c name , ' (' , (self getSize: c) asInteger printString , 'x)'
        ,
        (poly size > 0
        ifTrue: [ String cr , '---calls---' , String cr , (String cr join: poly) ]
        ifFalse: [ '' ]) ].
aView nodes: model allClasses.
aView edges: model allClasses from: [ :node | node ] to: [ :node | node
superclass ].
aView forceBasedLayout
ROEaselMorphic new populateMenuOn: view.
view openInWindowSized: windowSize.
```



visualisations demand expertise

We provide adaptable visualisation



Revisiting users need

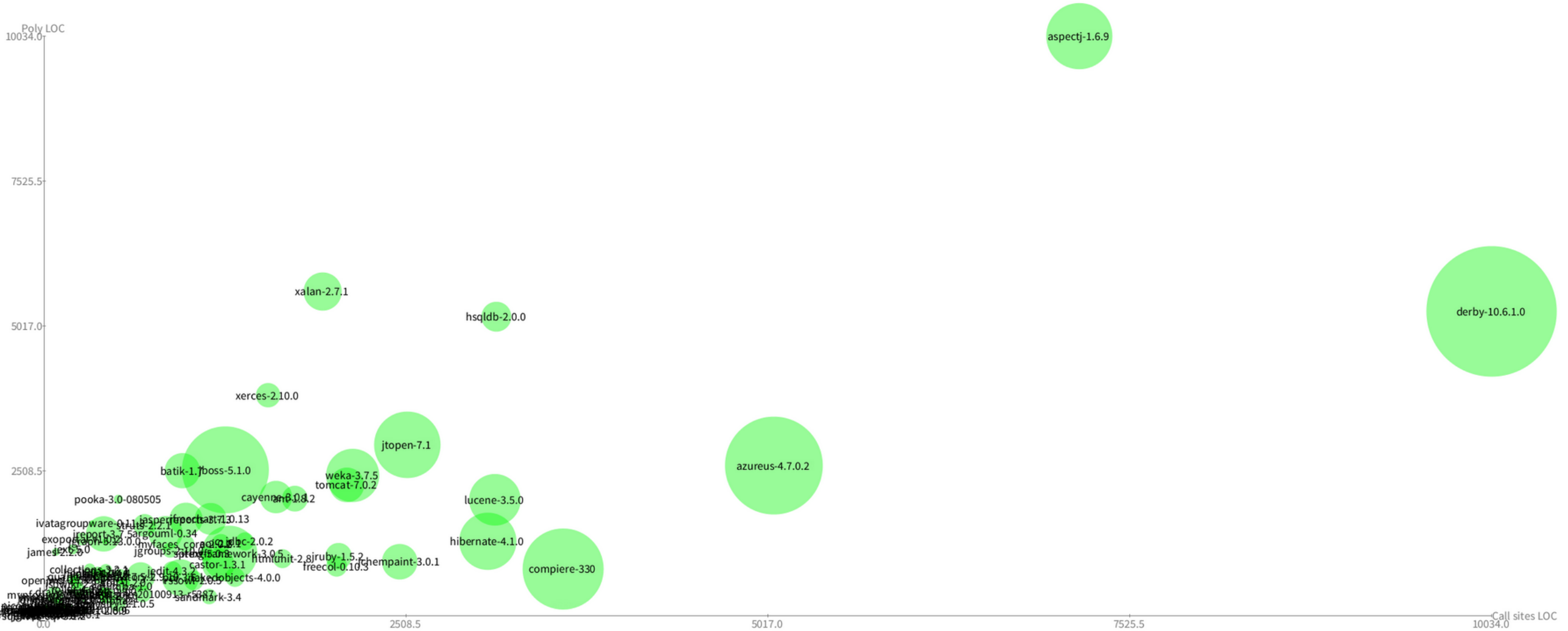
Use case: a researcher wants to analyse polymorphism and call sites at class level on a corpus.

- Which is the largest system?
- Which system has the most polymorphism?
- Which system has the most call sites?

DSL for specifying user needs

```
visualisationsDSL
    on: corpus;
    metric: heuristic;
    granularity:  class;
    keyword:  most.
```

A visualisation that fits this need

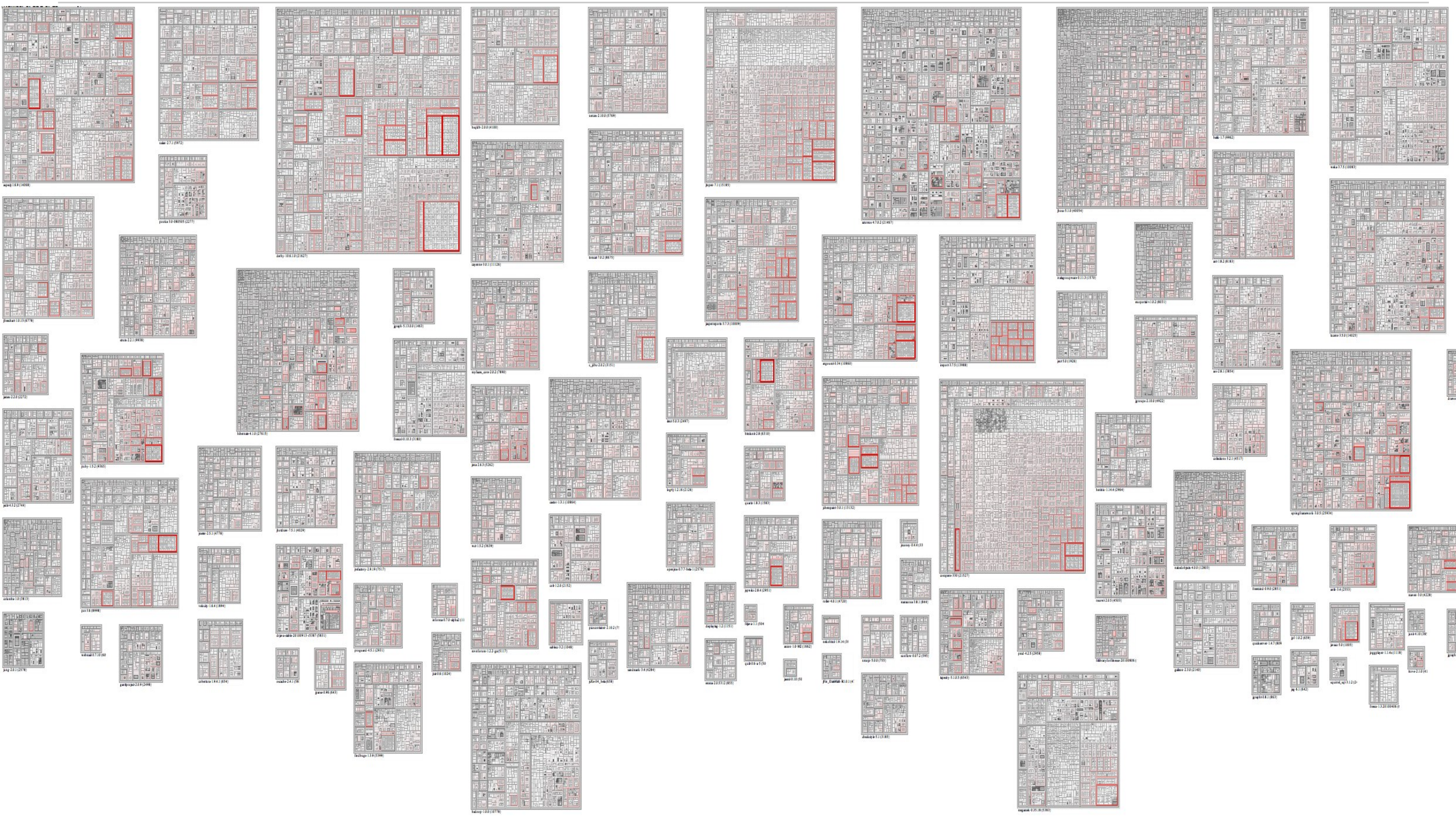


Another need

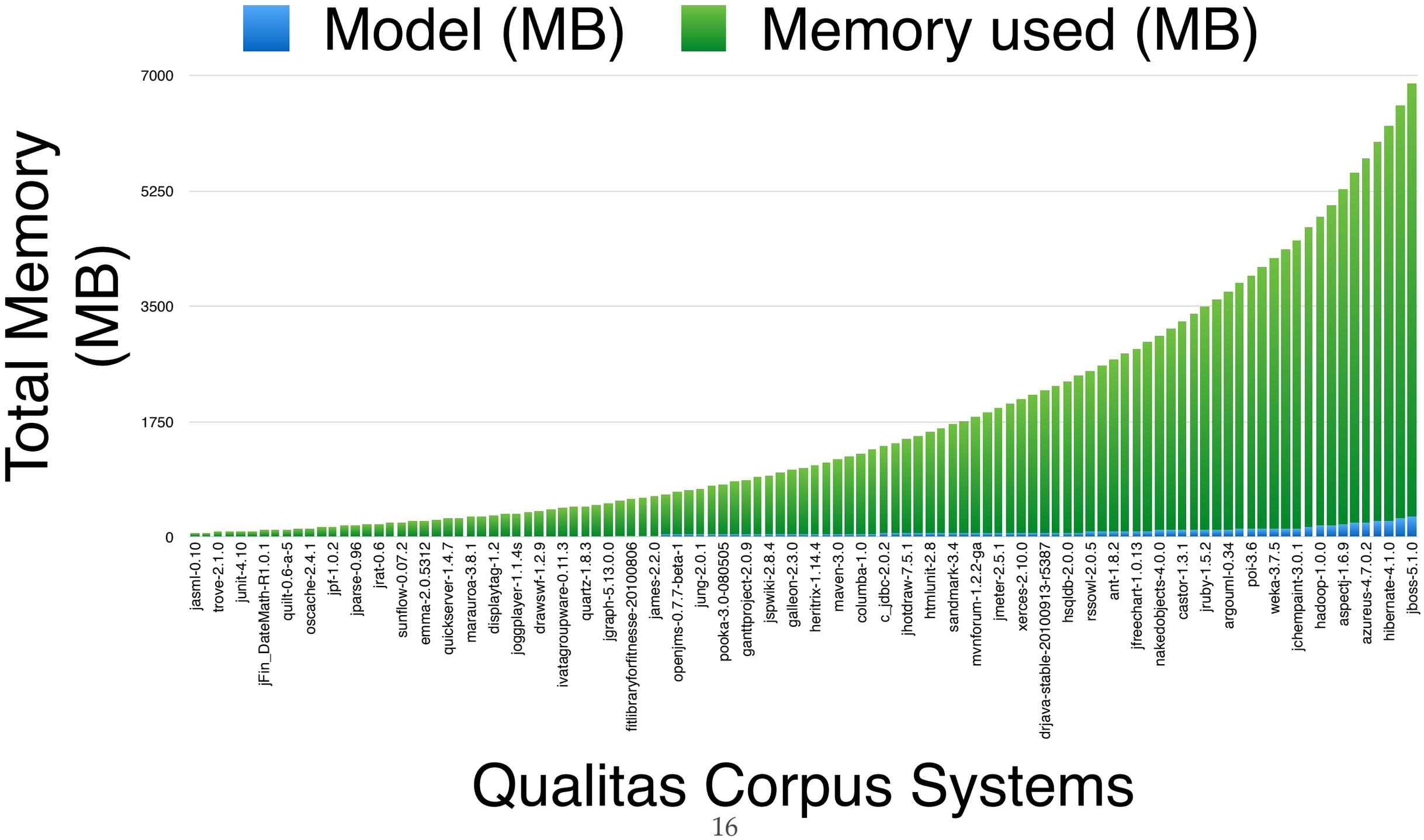
Use case: a researcher wants to analyse polymorphism and call sites at class level on a corpus.

- How polymorphism is distributed among classes?
- How call sites are distributed among classes?
- What about classes that presents polymorphism and call sites?

A visualisation that fits this need



Memory usage doesn't scale well



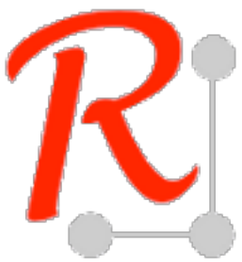
Scaling visualisations through D&C strategy

DIVIDE	ANALYSE	VISUALISE	COMPOSE
Dividing a large program into smaller blocks	Running analysis for sorting, colouring and sizing	Producing a visualisation of a block	Composing all visualisations together

Powered by

Pharo an object-oriented language

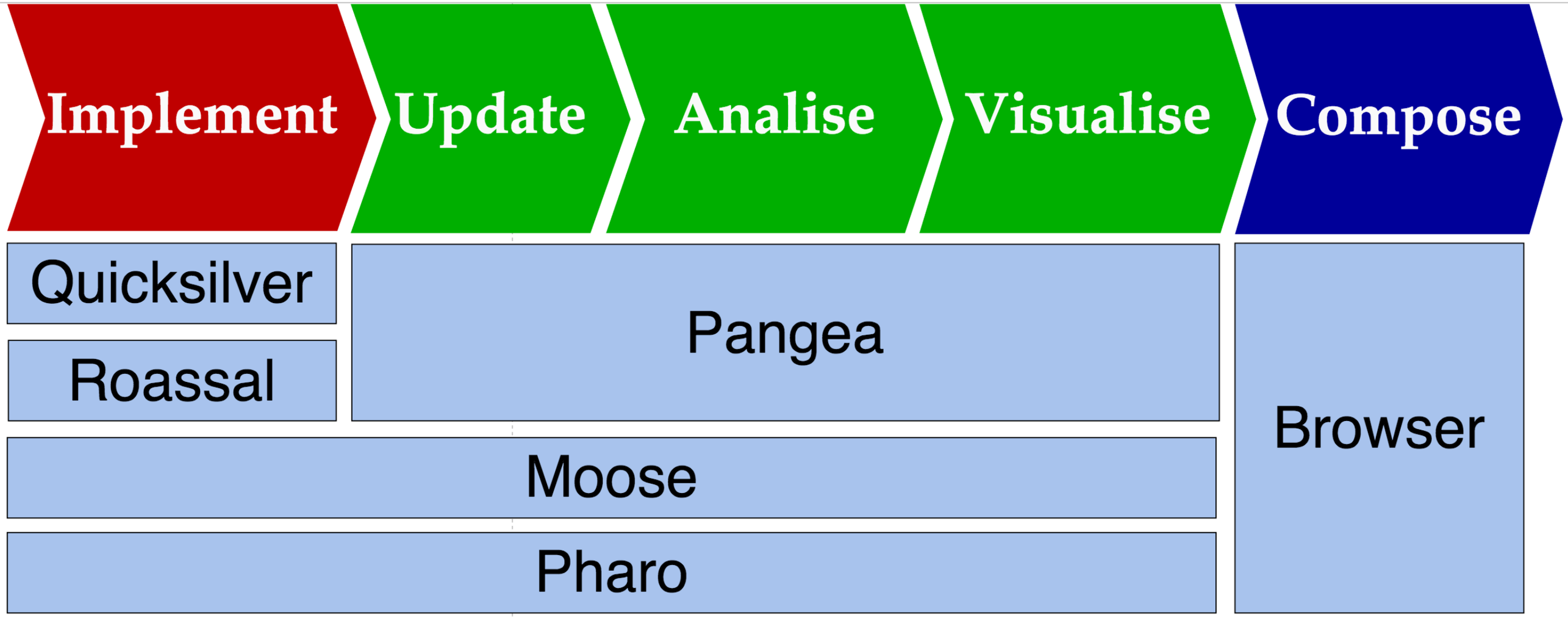
Moose a platform for software analysis



Roassal — Quicksilver visualisation engine

Pangea enables running language independent analysis on corpora of object-oriented software

D&C allowed us to scale



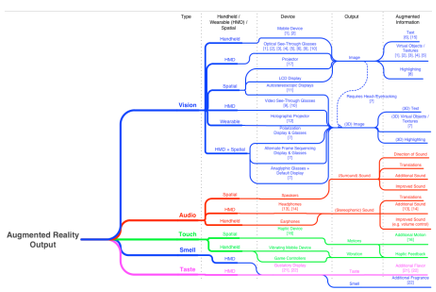
Qualitas Corpus (14’946 KLOC ~ 2 hrs.)

Summary:

Adaptable Visualisation based on user needs

Visualisations aid software analysis

demand expertise



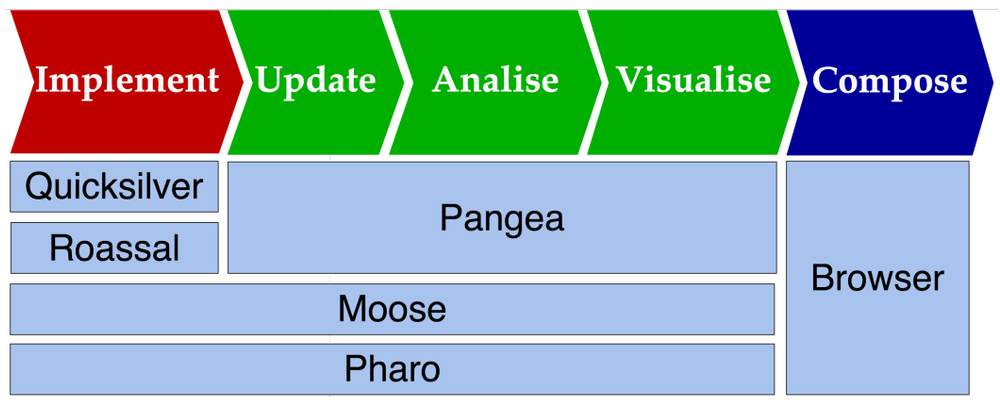
Survey

Taxonomy

DSL

```
visualisationsDSL
    on: corpus;
    metric: heuristic;
    granularity: class;
    keyword: most.
```

some don't scale well



Leonel Merino
merino @ iam.unibe.ch