Live Programming Seminar
Project Topics

Hasso Plattner Institute, University of Potsdam, Germany
Software Architecture Group
Robert Hirschfeld

Jens Lincke, Patrick Rein, Stefan Ramson, Tom Beckmann

http://www.hpi.uni-potsdam.de/swa/

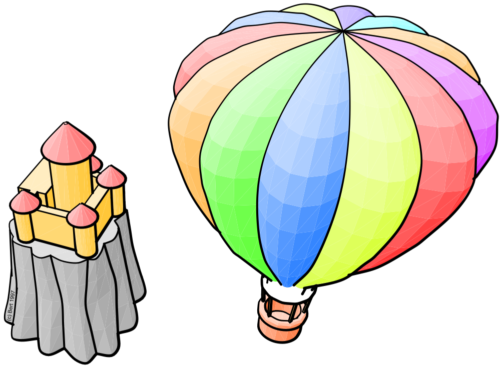
2021

First things first:

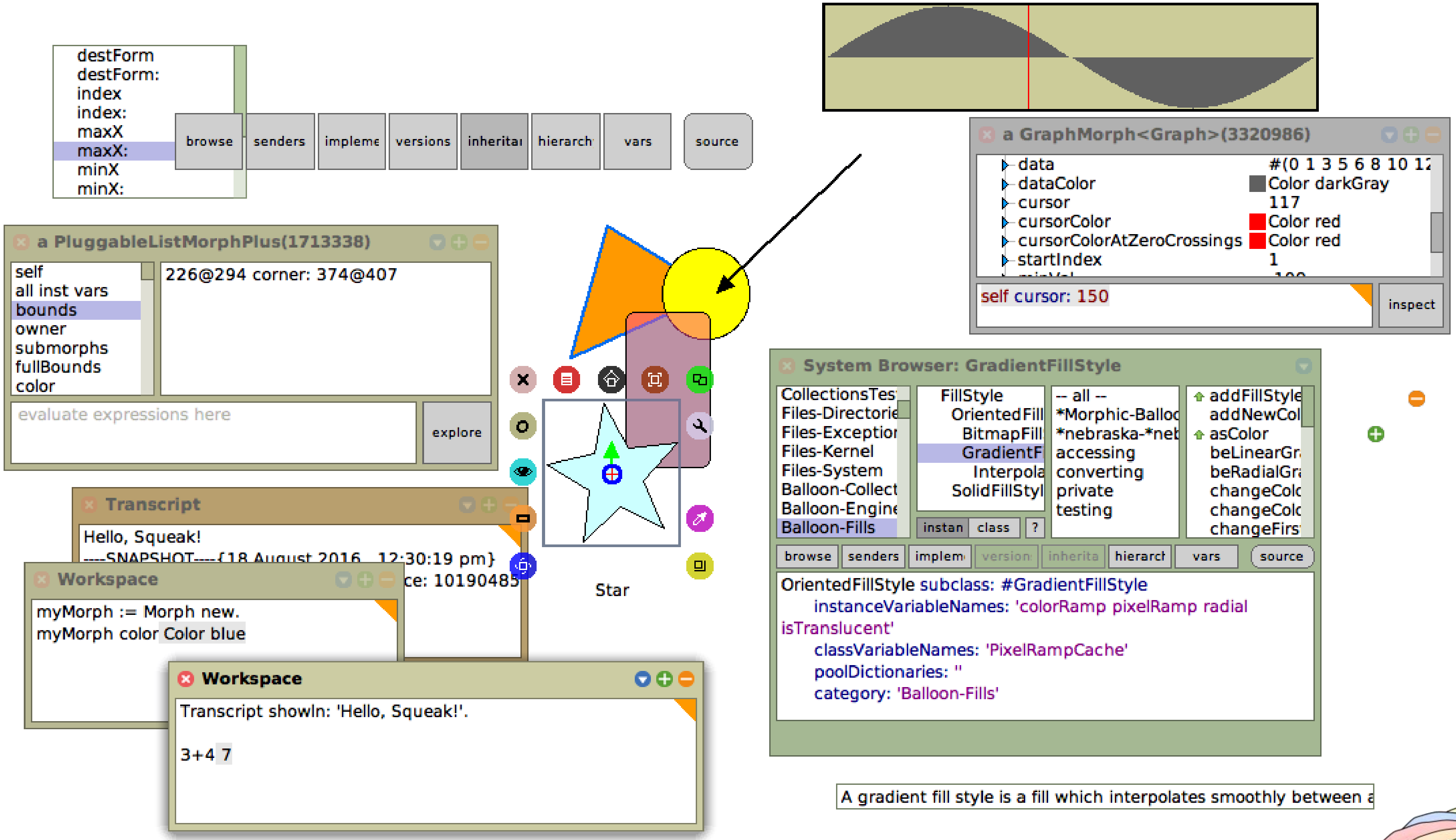
- Seminar Kickoff in Zoom, Thu April 15th, 13:30
  - Zoom Meeting https://uni-potsdam.zoom.us/j/69371244986
    Passcode: 78438809
- SWA Slack Channel for this Seminar #live21
  - https://join.slack.com/t/hpi-swa-teaching/shared_invite/zt-otx3xhv6-XSnQqjizT3gx0OKbJscsdw
  - Email jens.lincke@hpi.de

Development Environments

Lively Kernel, JavaScript



Squeak Smalltalk



```
226@294 corner: 374@407
```

```
Hello, Squeak!
----SNAPSHOT----{18 August 2016 . 12:30:19 pm}
ce: 10190485
```

```
myMorph := Morph new.
myMorph color Color blue
```

```
Transcript showln: 'Hello, Squeak!'.

3+4 7
```

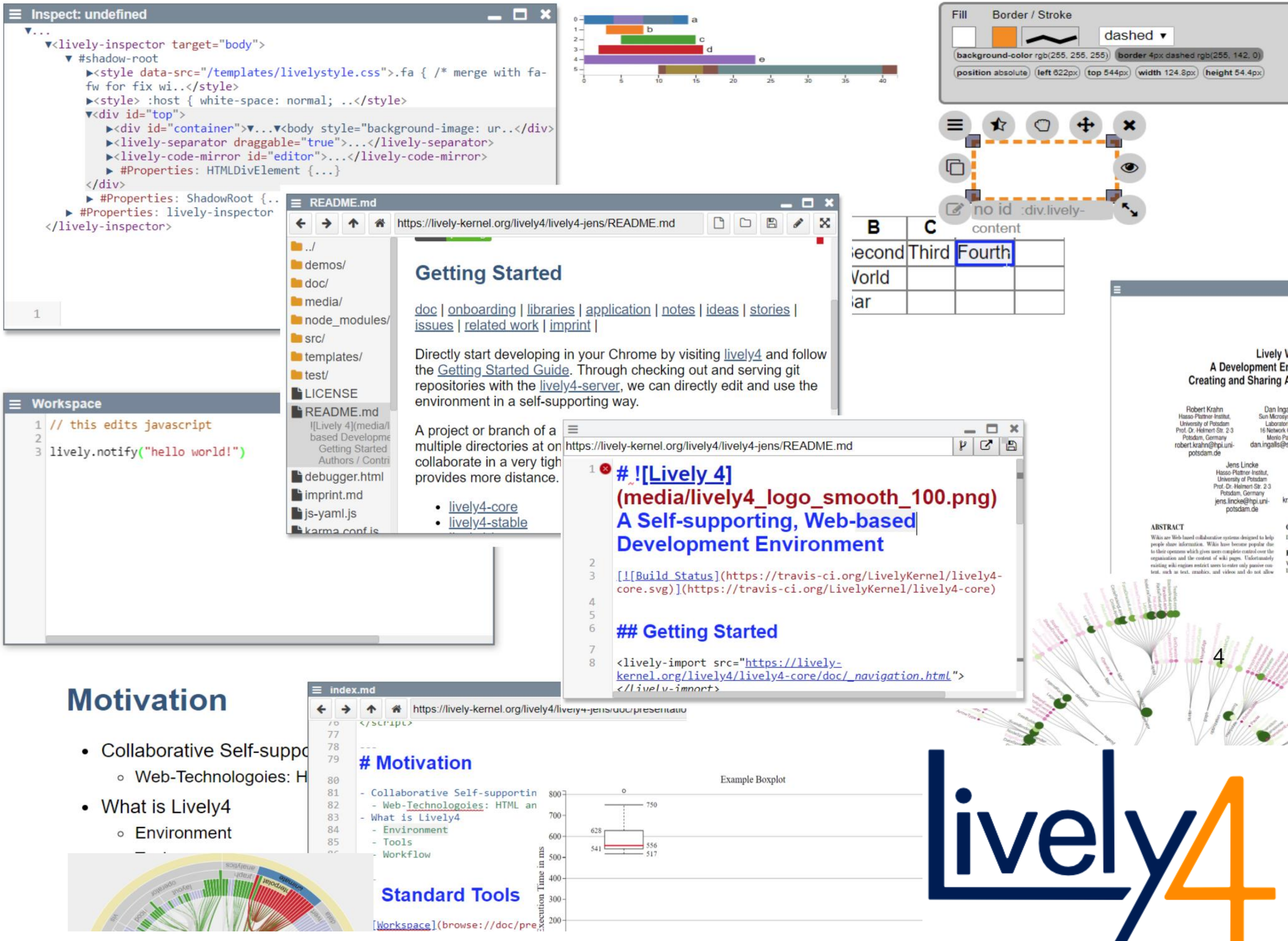
```
data                           #(0 1 3 5 6 8 10 12
dataColor                      ▇ Color darkGray
cursor                         117
cursorColor                    🟥 Color red
cursorColorAtZeroCrossings     🟥 Color red
startIndex                     1
```

```
self cursor: 150
```

```
OrientedFillStyle subclass: #GradientFillStyle
	instanceVariableNames: 'colorRamp pixelRamp radial
isTranslucent'
	classVariableNames: 'PixelRampCache'
	poolDictionaries: ''
	category: 'Balloon-Fills'
```

A gradient fill style is a fill which interpolates smoothly between a

Squeak/Smalltalk



```
▼...
  ▼<lively-inspector target="body">
    ▼ #shadow-root
      ▶<style data-src="/templates/livelystyle.css">.fa { /* merge with fa-
      fw for fix wi..</style>
      ▶<style> :host { white-space: normal; ..</style>
      ▼<div id="top">
        ▶<div id="container">▼...▼<body style="background-image: ur..</div>
        ▶<lively-separator draggable="true">...</lively-separator>
        ▶<lively-code-mirror id="editor">...</lively-code-mirror>
        ▶ #Properties: HTMLDivElement {...}
      </div>
      ▶ #Properties: ShadowRoot {..
    ▶ #Properties: lively-inspector
  </lively-inspector>
```

background-color rgb(255, 255, 255) | border 4px dashed rgb(255, 142, 0)

position absolute | left 622px | top 544px | width 124.8px | height 54.4px

B	C	
econd	Third	Fourth
Vorld		
ar		

https://lively-kernel.org/lively4/lively4-jens/README.md

Getting Started

doc | onboarding | libraries | application | notes | ideas | stories | issues | related work | imprint |

Directly start developing in your Chrome by visiting lively4 and follow the Getting Started Guide. Through checking out and serving git repositories with the lively4-server, we can directly edit and use the environment in a self-supporting way.

A project or branch of a
multiple directories at on
collaborate in a very tigh
provides more distance.

- lively4-core
- lively4-stable

https://lively-kernel.org/lively4/lively4-jens/README.md

```
# ![Lively 4](media/lively4_logo_smooth_100.png)
A Self-supporting, Web-based
Development Environment

[![Build Status](https://travis-ci.org/LivelyKernel/lively4-
core.svg)](https://travis-ci.org/LivelyKernel/lively4-core)

## Getting Started

<lively-import src="https://lively-
kernel.org/lively4/lively4-core/doc/_navigation.html">
</lively-import>
```

```
// this edits javascript

lively.notify("hello world!")
```

Lively W
A Development E
Creating and Sharing A

Robert Krahn
Hasso-Plattner-Institut,
University of Potsdam
Prof.-Dr.-Helmert-Str. 2-3
Potsdam, Germany
robert.krahn@hpi.uni-potsdam.de

Dan Inga
Sun Microsy
Laborator
16 Network
Menlo Pa
dan.ingalls@

Jens Lincke
Hasso-Plattner-Institut,
University of Potsdam
Prof.-Dr.-Helmert-Str. 2-3
Potsdam, Germany
jens.lincke@hpi.uni-potsdam.de

ABSTRACT

Wikis are Web-based collaborative systems designed to help people share information. Wikis have become popular due to their openness which gives users complete control over the organization and the content of wiki pages. Unfortunately existing wiki engines restrict users to enter only passive content, such as text, graphics, and videos and do not allow

Motivation

- Collaborative Self-suppo
  - Web-Technologoies: H
- What is Lively4
  - Environment

https://lively-kernel.org/lively4/lively4-jens/doc/presentatio

```
</script>

---
# Motivation

- Collaborative Self-supportin
  - Web-Technologoies: HTML an
- What is Lively4
  - Environment
  - Tools
  - Workflow
```

Standard Tools

[Workspace](browse://doc/pre

Software Architecture Group (www.hpi.uni-potsdam.de/swa) 2006-present

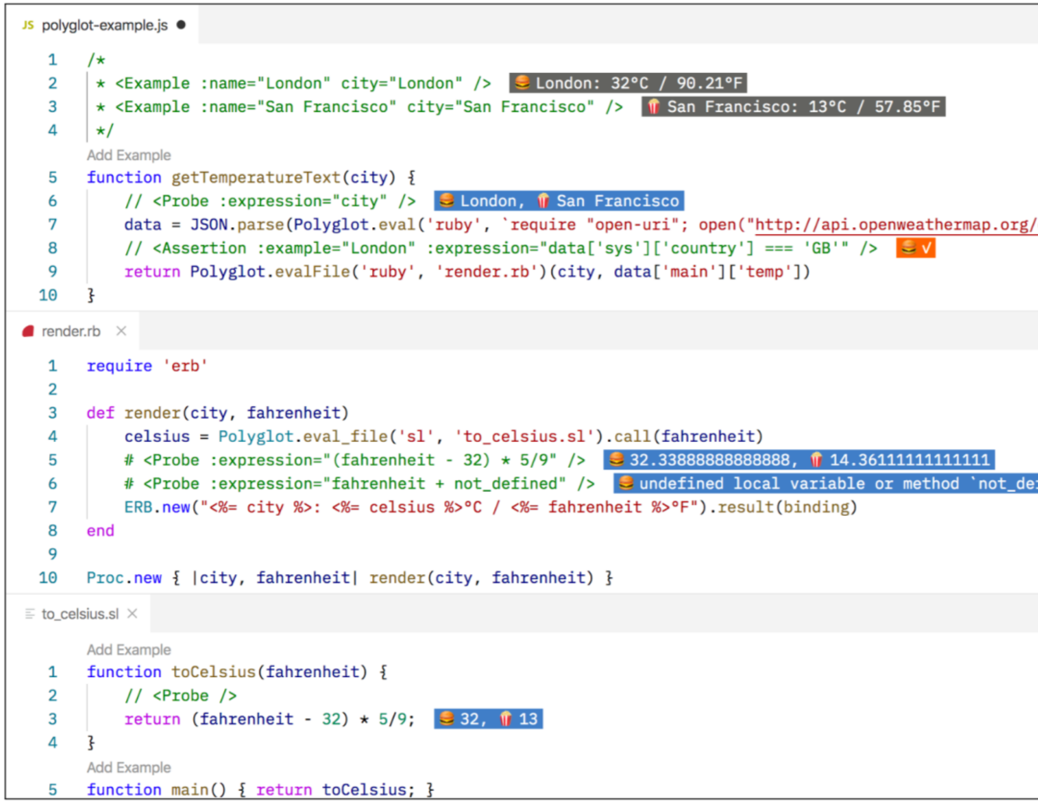
TOPICS

Topic: Babylonian Programming in IntelliJ

Bring Babylonian Programming to IntelliJ based on the polyglot live programming backend.

Challenges

- Design based on native IntelliJ Widgets
- Integration as plugin



polyglot-example.js ●

```
/*
 * <Example :name="London" city="London" />  🍔 London: 32°C / 90.21°F
 * <Example :name="San Francisco" city="San Francisco" />  🍿 San Francisco: 13°C / 57.85°F
 */
Add Example
function getTemperatureText(city) {
    // <Probe :expression="city" />  🍔 London, 🍿 San Francisco
    data = JSON.parse(Polyglot.eval('ruby', `require "open-uri"; open("http://api.openweathermap.org/
    // <Assertion :example="London" :expression="data['sys']['country'] === 'GB'" />  🍔 √
    return Polyglot.evalFile('ruby', 'render.rb')(city, data['main']['temp'])
}
```

render.rb ×

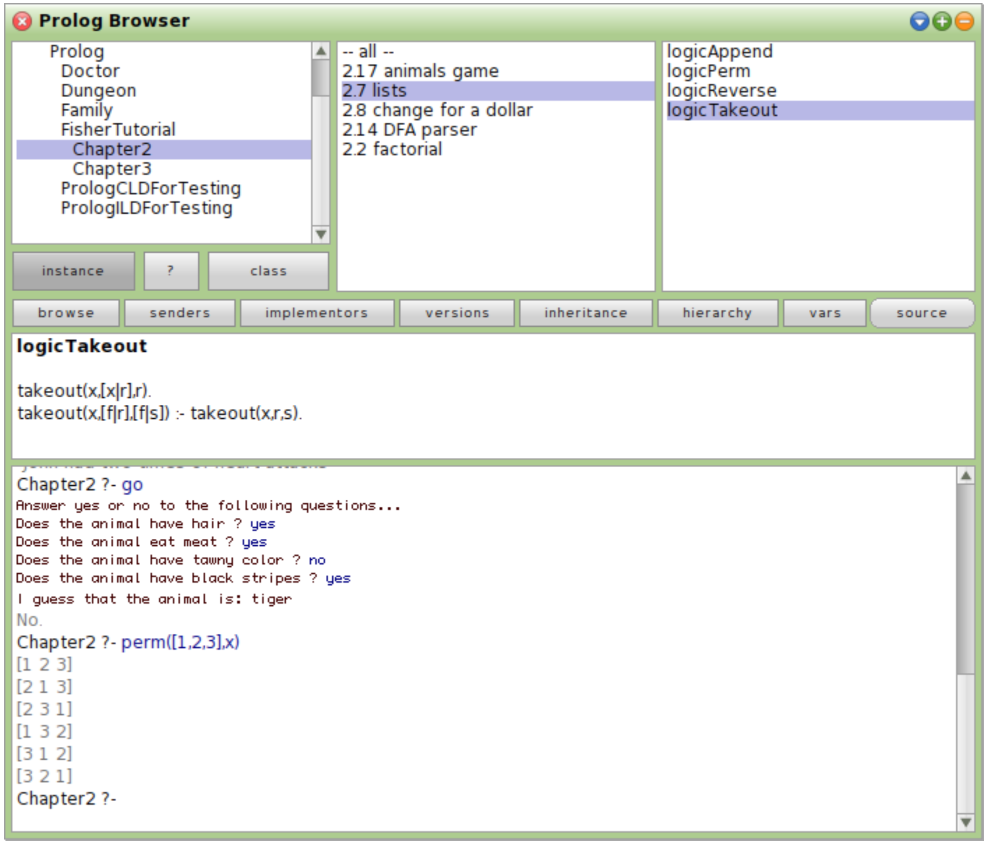
```
require 'erb'

def render(city, fahrenheit)
    celsius = Polyglot.eval_file('sl', 'to_celsius.sl').call(fahrenheit)
    # <Probe :expression="(fahrenheit - 32) * 5/9" />  🍔 32.33888888888888, 🍿 14.36111111111111
    # <Probe :expression="fahrenheit + not_defined" />  🍔 undefined local variable or method `not_de
    ERB.new("<%= city %>: <%= celsius %>°C / <%= fahrenheit %>°F").result(binding)
end

Proc.new { |city, fahrenheit| render(city, fahrenheit) }
```

to_celsius.sl ×

```
Add Example
function toCelsius(fahrenheit) {
    // <Probe />
    return (fahrenheit - 32) * 5/9;  🍔 32, 🍿 13
}
Add Example
function main() { return toCelsius; }
```

Topic: Babylonian Prolog

Design and implement a Babylonian-style programming environment for Prolog including examples and probes.



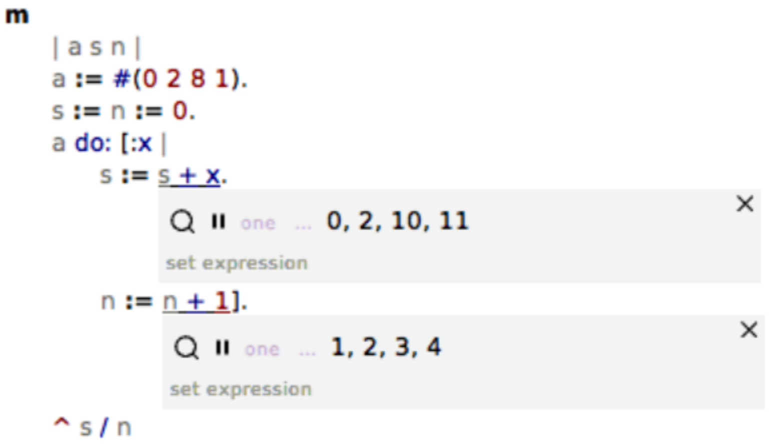
logicTakeout

```
takeout(x,[x|r],r).
takeout(x,[f|r],[f|s]) :- takeout(x,r,s).
```

```
Chapter2 ?- go
Answer yes or no to the following questions...
Does the animal have hair ? yes
Does the animal eat meat ? yes
Does the animal have tawny color ? no
Does the animal have black stripes ? yes
I guess that the animal is: tiger
No.
Chapter2 ?- perm([1,2,3],x)
[1 2 3]
[2 1 3]
[2 3 1]
[1 3 2]
[3 1 2]
[3 2 1]
Chapter2 ?-
```

Topic: Live Printf

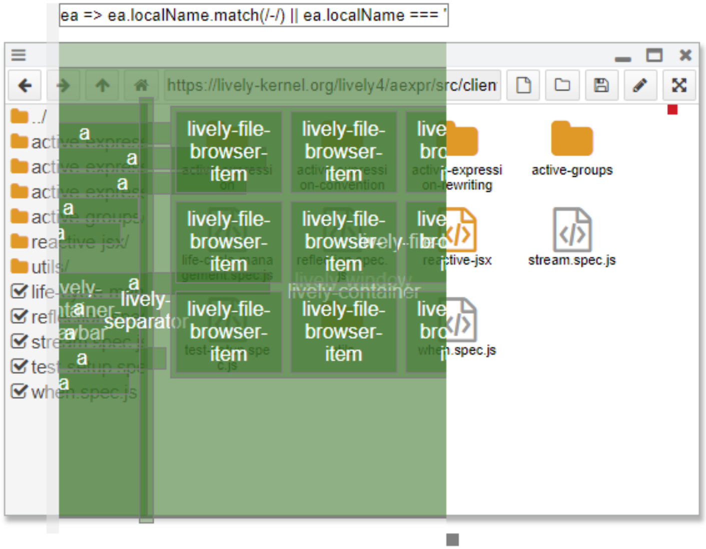
Explore and implement a live view of run-time behavior similar to a printf-log but based on the results of probes.



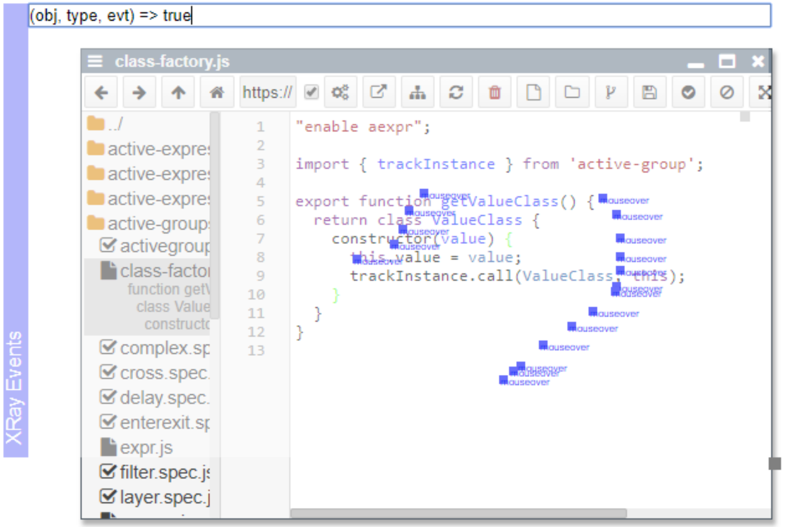
```
m
    | a s n |
    a := #(0 2 8 1).
    s := n := 0.
    a do: [:x |
        s := s + x.
            one ... 0, 2, 10, 11
            set expression
        n := n + 1].
            one ... 1, 2, 3, 4
            set expression
    ^ s / n
```

Motivation
Probes provide live, over-time feedback on the results of an expression, printf-logging provides feedback on the temporal succession of expressions throughout the program.

Context: X Ray



ea => ea.localName.match(/-/) || ea.localName === '



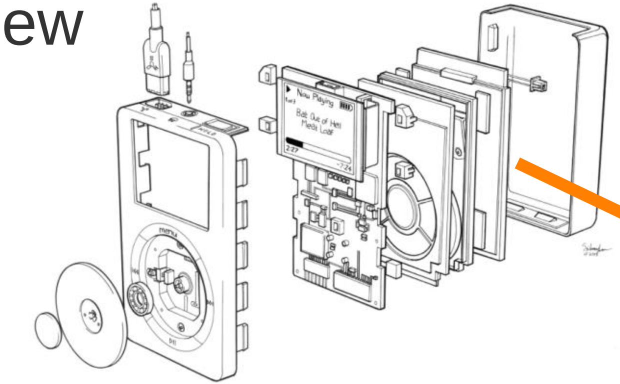
(obj, type, evt) => true

```
"enable aexpr";

import { trackInstance } from 'active-group';

export function getValueClass() {
  return class ValueClass {
    constructor(value) {
      this.value = value;
      trackInstance.call(ValueClass, this);
    }
  }
}
```

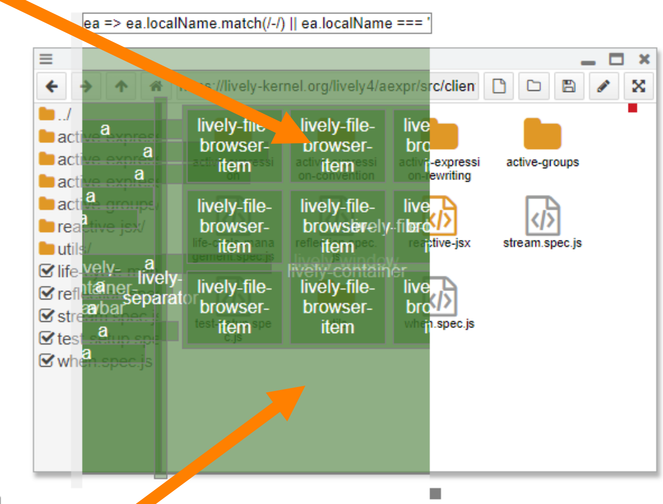
- Lively XRay tools overlay graphical user interface to:
  - Reveal and inspect structure of the DOM hierarchy
  - Make user events visible

Topic: X Ray (A) Better UI

- Explosion View

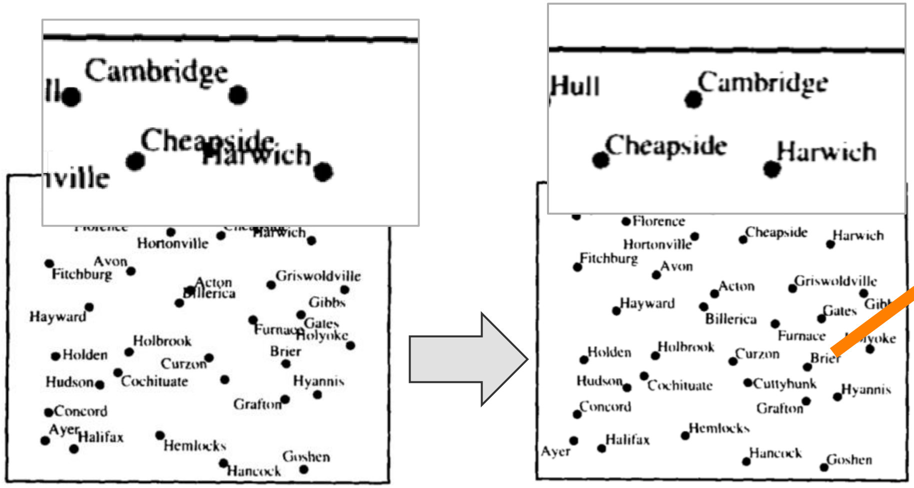


[source: https://needmoredesigns.com/on-building-from-scratch/]



ea => ea.localName.match(/-/) || ea.localName === '
https://lively-kernel.org/lively4/aexpr/src/clien
../
utils/
lively-file-browser-item
lively-container
lively-separator
active-groups
active-expressi on-rewriting
reactive-jsx
stream.spec.js
when.spec.js

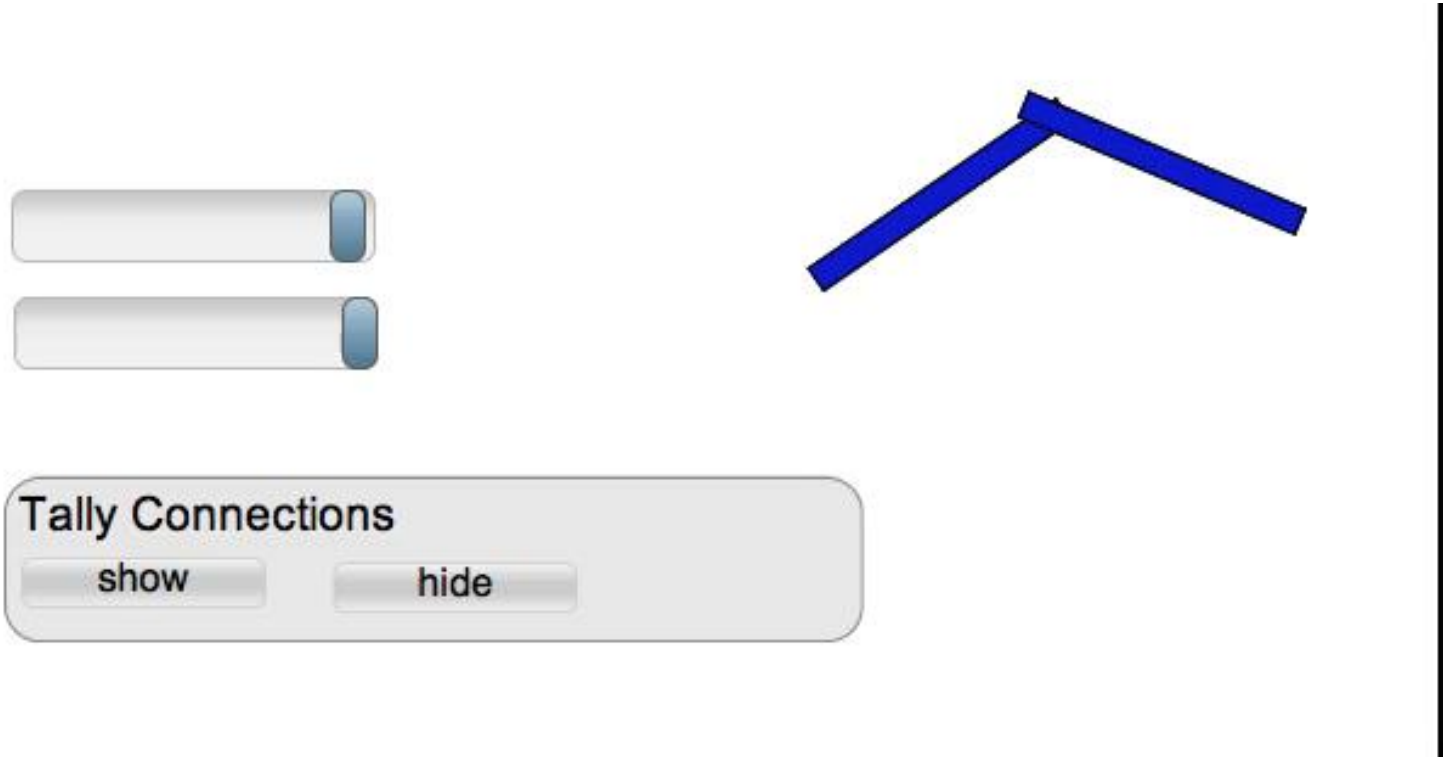
- Better Label Placement



[Christensen 1995. An Empirical Study of Algorithms for Point-Feature Label Placement]

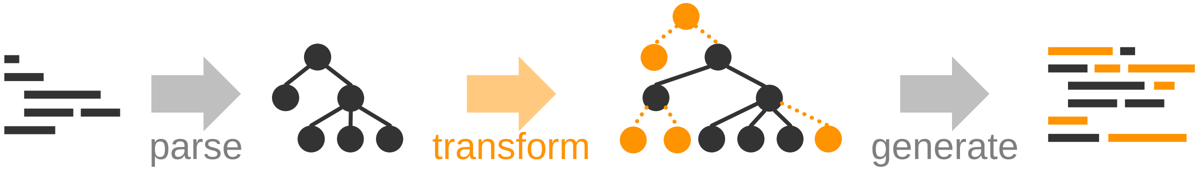
Topic: X Ray (B) Show (Async) Behavior

- **Goal:** Extend XRay tools to reveal behavior
  - Trace and show computation within components
  - Link computation to dynamic DOM structure and static source code
  - Highlight reactive components and their over-time behavior



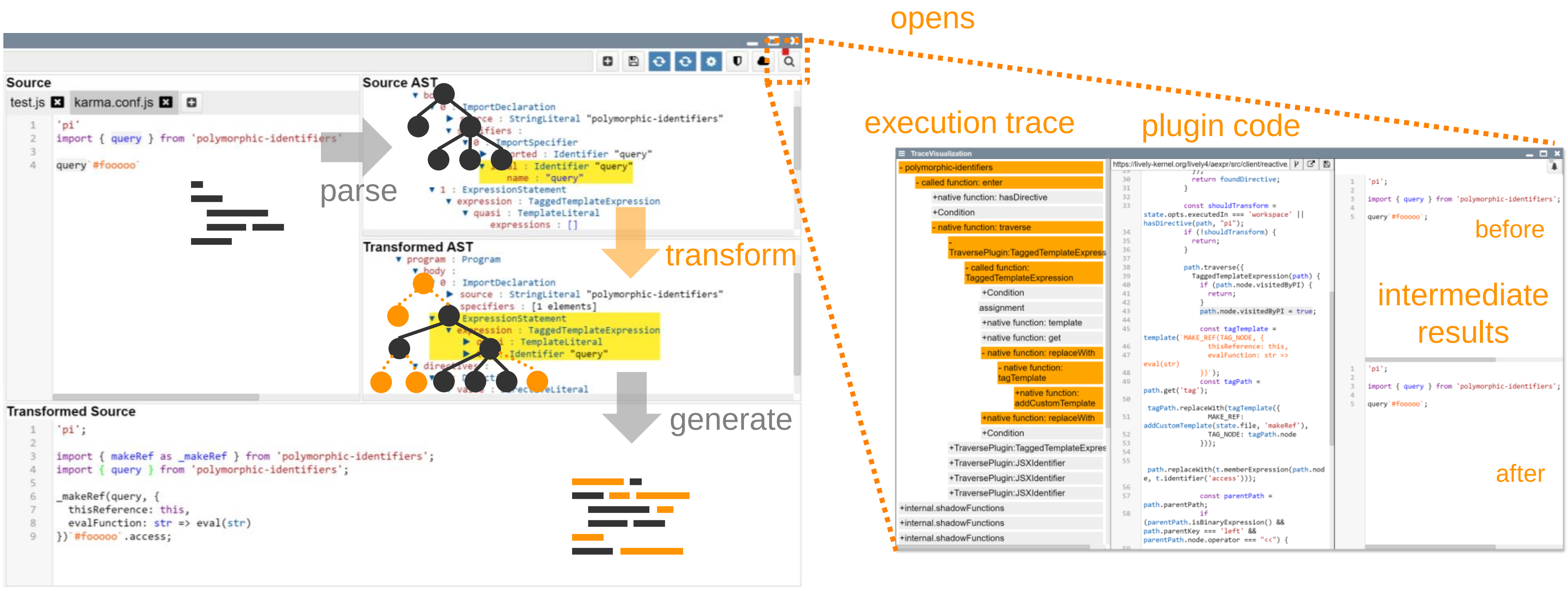
Topic: Live Programming with Back-in-Time Debugging for Source Code Transformations (1/3)

- Context



  - Source code transformations are an integral part in JavaScript workflows to support various language **dialects**, custom **extensions**, and future language **proposals**
  - Multiple interacting transformations entail **high complexity**
- Opportunity
  - A **dedicated tracing tool** may reveal why rules trigger, what effects a rule has, and which further rules apply
  - Connecting input code, output code, and transformations allow for navigating examples **forward and backward in time**
- But
  - Loss of fast feedback loop hampers programming experience
  - **Separation** of tools **discourages adoption** of less frequently-used tools

Topic: Live Programming with Back-in-Time Debugging for Source Code Transformations (2/3)



Source

test.js ☒ karma.conf.js ☒

```
'pi'
import { query } from 'polymorphic-identifiers'

query`#fooooo`
```

Source AST

```
▼ bo
  ▼ 0 : ImportDeclaration
    ▶ source : StringLiteral "polymorphic-identifiers"
    ▼ specifiers :
      ▼ 0 : ImportSpecifier
        ▶ imported : Identifier "query"
        ▼ local : Identifier "query"
            name : "query"
  ▼ 1 : ExpressionStatement
    ▼ expression : TaggedTemplateExpression
      ▼ quasi : TemplateLiteral
          expressions : []
```

Transformed AST

```
▼ program : Program
  ▼ body :
    ▼ 0 : ImportDeclaration
      ▶ source : StringLiteral "polymorphic-identifiers"
        specifiers : [1 elements]
    ▼ ExpressionStatement
      ▼ expression : TaggedTemplateExpression
        ▶ quasi : TemplateLiteral
        ▶ Identifier "query"
  ▼ directives :
      value : DirectiveLiteral
```

Transformed Source

```
'pi';

import { makeRef as _makeRef } from 'polymorphic-identifiers';
import { query } from 'polymorphic-identifiers';

_makeRef(query, {
  thisReference: this,
  evalFunction: str => eval(str)
})`#fooooo`.access;
```

```
- polymorphic-identifiers
  - called function: enter
    +native function: hasDirective
    +Condition
    - native function: traverse
      - TraversePlugin:TaggedTemplateExpress
        - called function: TaggedTemplateExpression
          +Condition
          assignment
          +native function: template
          +native function: get
          - native function: replaceWith
            - native function: tagTemplate
              +native function: addCustomTemplate
          +native function: replaceWith
          +Condition
      +TraversePlugin:TaggedTemplateExpres
      +TraversePlugin:JSXIdentifier
      +TraversePlugin:JSXIdentifier
      +TraversePlugin:JSXIdentifier
+internal.shadowFunctions
+internal.shadowFunctions
+internal.shadowFunctions
```

https://lively-kernel.org/lively4/aexpr/src/client/reactive

```
          return foundDirective;
        }

        const shouldTransform =
state.opts.executedIn === 'workspace' ||
hasDirective(path, "pi");
        if (!shouldTransform) {
          return;
        }

        path.traverse({
          TaggedTemplateExpression(path) {
            if (path.node.visitedByPI) {
              return;
            }
            path.node.visitedByPI = true;

            const tagTemplate =
template(`MAKE_REF(TAG_NODE, {
              thisReference: this,
              evalFunction: str =>
eval(str)
            })`);
            const tagPath =
path.get('tag');

 tagPath.replaceWith(tagTemplate({
              MAKE_REF:
addCustomTemplate(state.file, 'makeRef'),
              TAG_NODE: tagPath.node
            }));


 path.replaceWith(t.memberExpression(path.nod
e, t.identifier('access')));

            const parentPath =
path.parentPath;
            if
(parentPath.isBinaryExpression() &&
path.parentKey === 'left' &&
parentPath.node.operator === "<<") {
```

```
'pi';

import { query } from 'polymorphic-identifiers';

query`#fooooo`;
```

```
'pi';

import { query } from 'polymorphic-identifiers';

query`#fooooo`;
```

Live Plugin Explorer

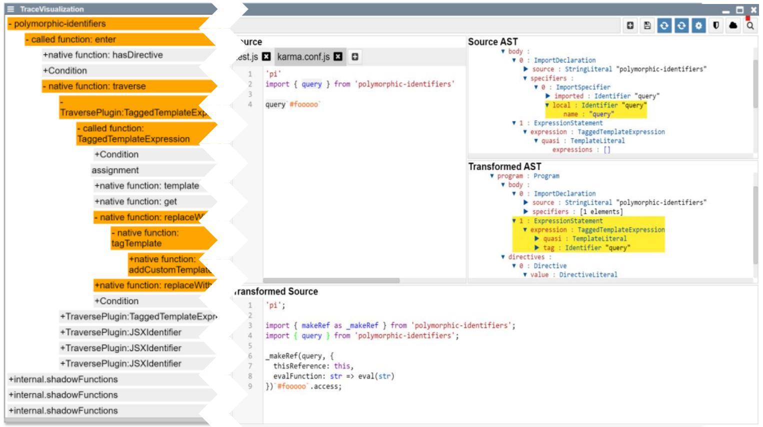
programming transformations

Trace Visualization

navigate execution history

Topic: Live Programming with Back-in-Time Debugging for Source Code Transformations (3/3)

- Goal
  - **Integrate both** tools to shorten feedback loop on powerful insights


TraceVisualization
- polymorphic-identifiers
- called function: enter
+native function: hasDirective
+Condition
- native function: traverse
- TraversePlugin:TaggedTemplateExp
- called function: TaggedTemplateExpression
+Condition
assignment
+native function: template
+native function: get
- native function: replaceW
- native function: tagTemplate
+native function: addCustomTemplat
+native function: replaceWith
+Condition
+TraversePlugin:TaggedTemplateExpr
+TraversePlugin:JSXIdentifier
+TraversePlugin:JSXIdentifier
+TraversePlugin:JSXIdentifier
+internal.shadowFunctions
+internal.shadowFunctions
+internal.shadowFunctions
urce
est.js
karma.conf.js

```
'pi'
import { query } from 'polymorphic-identifiers'

query`#fooooo`
```

Source AST

```
▼ body :
  ▼ 0 : ImportDeclaration
    ▶ source : StringLiteral "polymorphic-identifiers"
    ▼ specifiers :
      ▼ 0 : ImportSpecifier
        ▶ imported : Identifier "query"
        ▼ local : Identifier "query"
            name : "query"
  ▼ 1 : ExpressionStatement
    ▼ expression : TaggedTemplateExpression
      ▼ quasi : TemplateLiteral
          expressions : []
```

Transformed AST

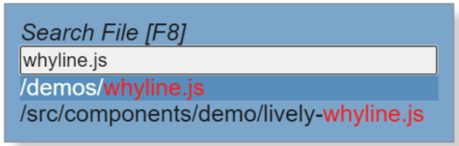
```
▼ program : Program
  ▼ body :
    ▼ 0 : ImportDeclaration
      ▶ source : StringLiteral "polymorphic-identifiers"
      ▶ specifiers : [1 elements]
    ▼ 1 : ExpressionStatement
      ▼ expression : TaggedTemplateExpression
        ▶ quasi : TemplateLiteral
        ▶ tag : Identifier "query"
  ▼ directives :
    ▼ 0 : Directive
      ▼ value : DirectiveLiteral
```

ransformed Source

```
'pi';

import { makeRef as _makeRef } from 'polymorphic-identifiers';
import { query } from 'polymorphic-identifiers';

_makeRef(query, {
  thisReference: this,
  evalFunction: str => eval(str)
})`#fooooo`.access;
```

- Readings
  - Whyline
  - Inventing on Principle
  - Babylonian-style Programming + Babylonian Demos in Lively4
  - Moldable Debugger
- Getting started
  - Go to https://lively-kernel.org/lively4/lively4-core/start.html

F8 to search



Search File [F8]

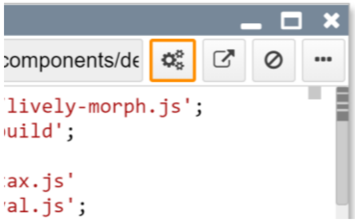
whyline.js

/demos/whyline.js

/src/components/demo/lively-whyline.js

plugin-explorer.js

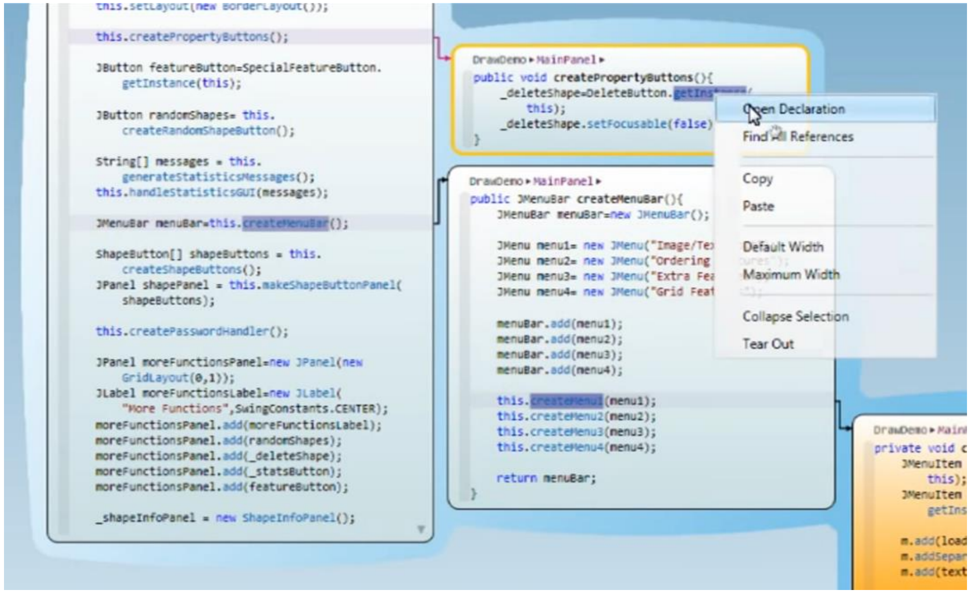
open an example



```
lively-morph.js';
uild';

ax.js'
al.js';
```

Topic: Spatial Arrangement of Software Systems

Explore and implement manners to manually curate spatial layouts of large code bases on a 2D grid in a block-based programming environment

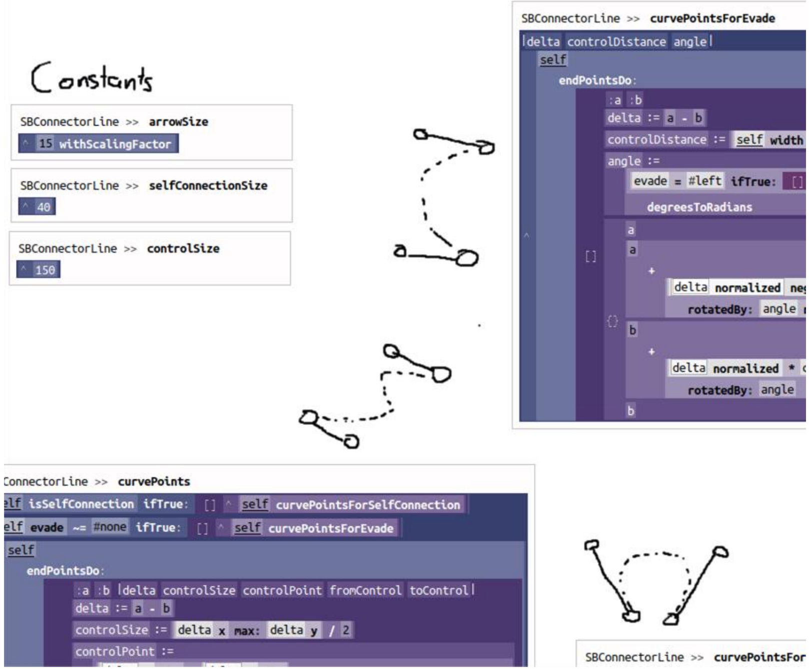


```
this.setLayout(new BorderLayout());

this.createPropertyButtons();

JButton featureButton=SpecialFeatureButton.
    getInstance(this);

JButton randomShapes= this.
    createRandomShapeButton();

String[] messages = this.
    generateStatisticsMessages();
this.handleStatisticsGUI(messages);

JMenuBar menuBar=this.createMenuBar();

ShapeButton[] shapeButtons = this.
    createShapeButtons();
JPanel shapePanel = this.makeShapeButtonPanel(
    shapeButtons);

this.createPasswordHandler();

JPanel moreFunctionsPanel=new JPanel(new
    GridLayout(0,1));
JLabel moreFunctionsLabel=new JLabel(
    "More Functions",SwingConstants.CENTER);
moreFunctionsPanel.add(moreFunctionsLabel);
moreFunctionsPanel.add(randomShapes);
moreFunctionsPanel.add(_deleteShape);
moreFunctionsPanel.add(_statsButton);
moreFunctionsPanel.add(featureButton);

_shapeInfoPanel = new ShapeInfoPanel();
```

```
DrawDemo ▸ MainPanel ▸
public void createPropertyButtons(){
    _deleteShape=DeleteButton.getIns
        this);
    _deleteShape.setFocusable(false)
}
```

```
DrawDemo ▸ MainPanel ▸
public JMenuBar createMenuBar(){
    JMenuBar menuBar=new JMenuBar();

    JMenu menu1= new JMenu("Image/Tex
    JMenu menu2= new JMenu("Ordering
    JMenu menu3= new JMenu("Extra Fea
    JMenu menu4= new JMenu("Grid Feat

    menuBar.add(menu1);
    menuBar.add(menu2);
    menuBar.add(menu3);
    menuBar.add(menu4);

    this.createMenu1(menu1);
    this.createMenu2(menu2);
    this.createMenu3(menu3);
    this.createMenu4(menu4);

    return menuBar;
}
```

```
DrawDemo ▸ Main
private void c
    JMenuIten
        this);
    JMenuIten
        getIns

    m.add(load
    m.addSepar
    m.add(text
```



```
SBConnectorLine >> arrowSize
^ 15 withScalingFactor
```

```
SBConnectorLine >> selfConnectionSize
^ 40
```

```
SBConnectorLine >> controlSize
^ 150
```

```
SBConnectorLine >> curvePointsForEvade
|delta controlDistance angle|
^ self
    endPointsDo:
        [:a :b
        delta := a - b
        controlDistance := self width
        angle :=
            evade = #left ifTrue: []
                degreesToRadians
        {a
         a
            +
                delta normalized neg
                    rotatedBy: angle
         b
            +
                delta normalized *
                    rotatedBy: angle
         b}
```

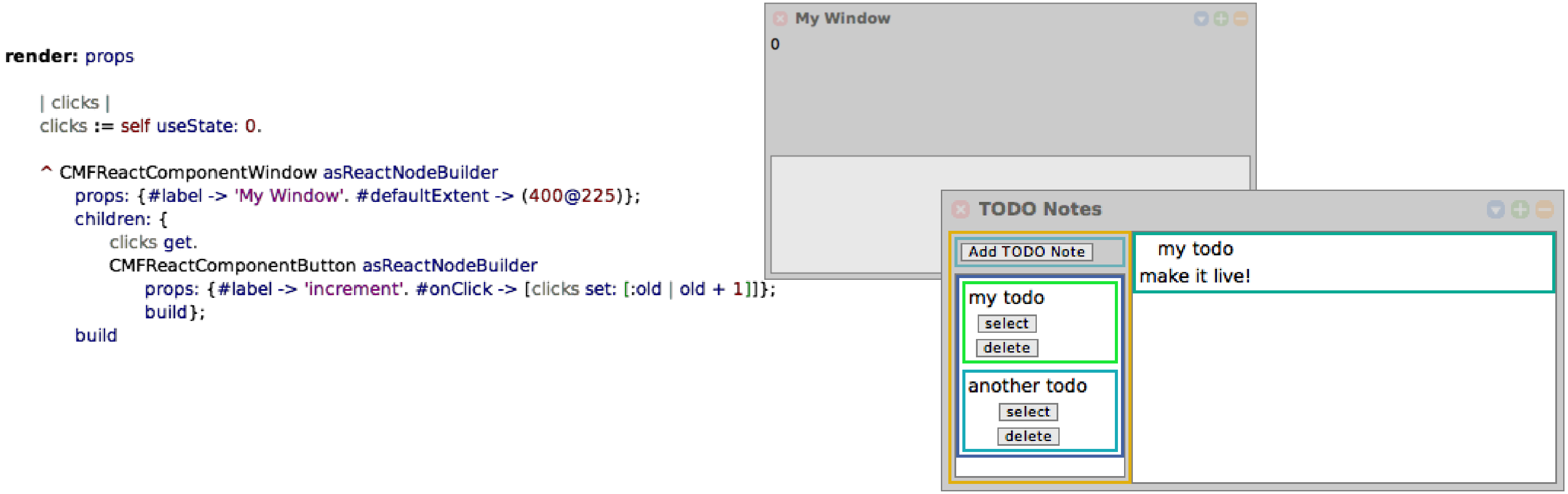
```
ConnectorLine >> curvePoints
elf isSelfConnection ifTrue: [] ^ self curvePointsForSelfConnection
elf evade ~= #none ifTrue: [] ^ self curvePointsForEvade
self
    endPointsDo:
        :a :b |delta controlSize controlPoint fromControl toControl|
        delta := a - b
        controlSize := delta x max: delta y / 2
        controlPoint :=
```

```
SBConnectorLine >> curvePointsFor
```

Software Architecture Group (www.hpi.uni-potsdam.de/swa) 2006-present

Topic: React/S

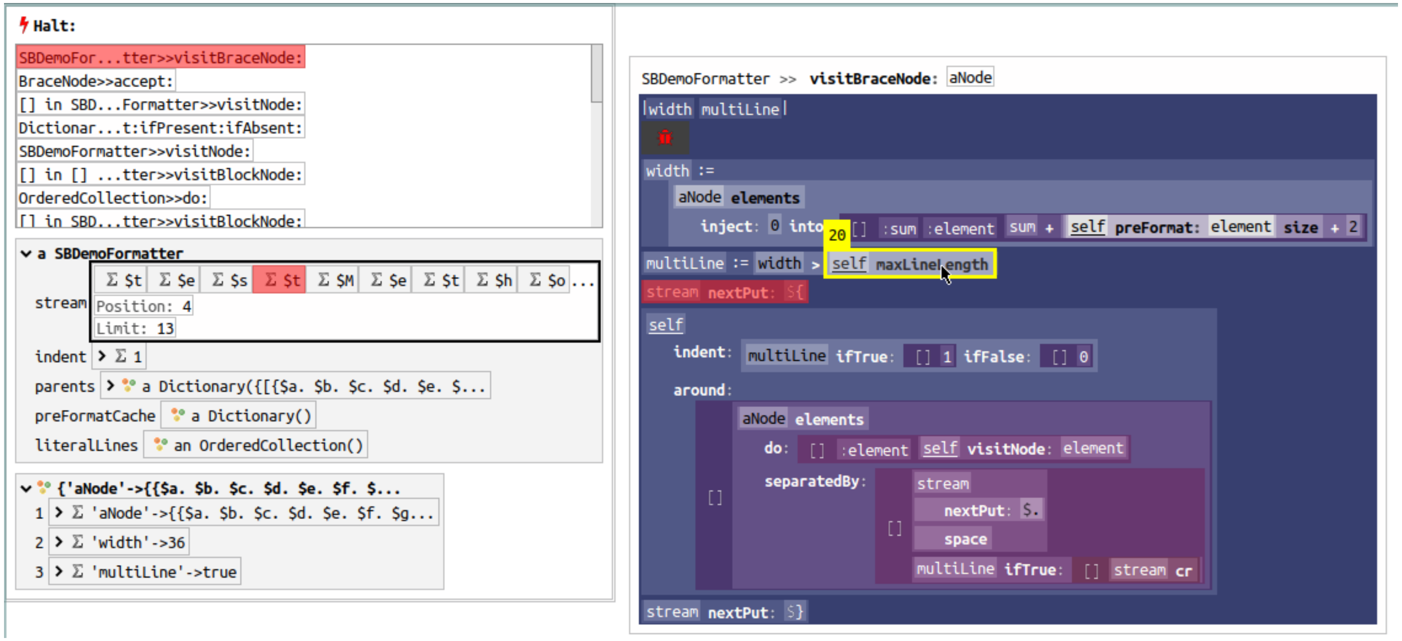
Extend and build tool support for a React implementation in Squeak with a focus on liveness and explorability.



```
render: props

    | clicks |
    clicks := self useState: 0.

    ^ CMFReactComponentWindow asReactNodeBuilder
        props: {#label -> 'My Window'. #defaultExtent -> (400@225)};
        children: {
            clicks get.
            CMFReactComponentButton asReactNodeBuilder
                props: {#label -> 'increment'. #onClick -> [clicks set: [:old | old + 1]]};
                build};
        build
```

Topic: Debugging with Blocks

Explore and implement visualizations in a block-based programming environment connected to a stepwise debugger.

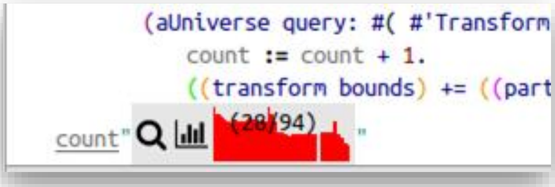


```
Halt:
SBDemoFor...tter>>visitBraceNode:
BraceNode>>accept:
[] in SBD...Formatter>>visitNode:
Dictionar...t:ifPresent:ifAbsent:
SBDemoFormatter>>visitNode:
[] in [] ...tter>>visitBlockNode:
OrderedCollection>>do:
[] in SBD...tter>>visitBlockNode:

⌄ a SBDemoFormatter
    stream  Σ $t  Σ $e  Σ $s  Σ $t  Σ $M  Σ $e  Σ $t  Σ $h  Σ $o ...
            Position: 4
            Limit: 13
    indent  > Σ 1
    parents  > a Dictionary({[{$a. $b. $c. $d. $e. $...
    preFormatCache  a Dictionary()
    literalLines  an OrderedCollection()

⌄ {'aNode'->{{$a. $b. $c. $d. $e. $f. $...
  1 > Σ 'aNode'->{{$a. $b. $c. $d. $e. $f. $g...
  2 > Σ 'width'->36
  3 > Σ 'multiLine'->true
```

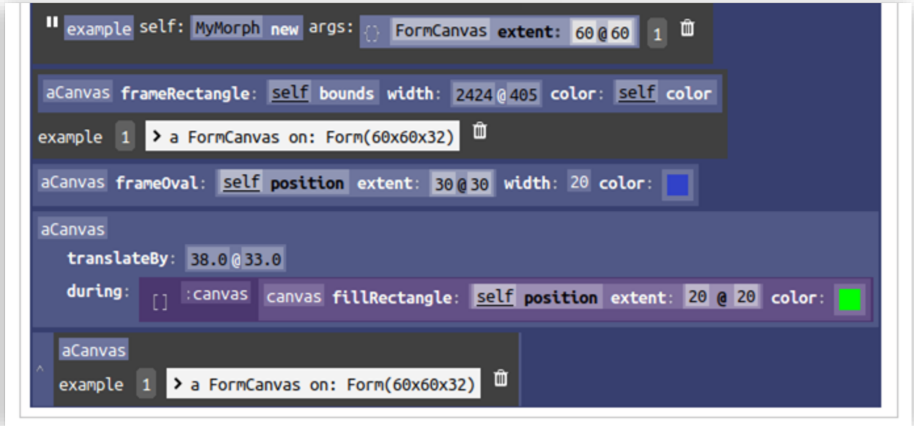
```
SBDemoFormatter >> visitBraceNode: aNode
|width multiLine|

width :=
    aNode elements
      inject: 0 into: [] :sum :element sum + self preFormat: element size + 2
multiLine := width > self maxLineLength   20
stream nextPut: ${
self
  indent: multiLine ifTrue: [] 1 ifFalse: [] 0
  around:
  [] aNode elements
       do: [] :element self visitNode: element
       separatedBy: [] stream
                          nextPut: $.
                          space
                       multiLine ifTrue: [] stream cr
stream nextPut: $}
```

Topic: Runtime Value Visualization

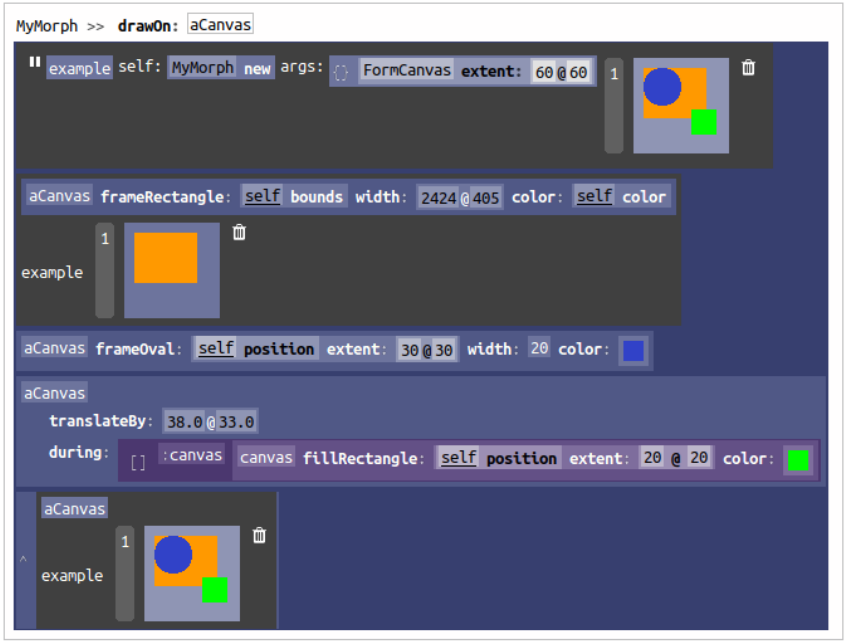
Explore and implement rich, domain-specific visualizations that display and/or accumulate runtime values in probes.



```
(aUniverse query: #( #'Transform
    count := count + 1.
    ((transform bounds) += ((part
count"      (28/94)  "
```



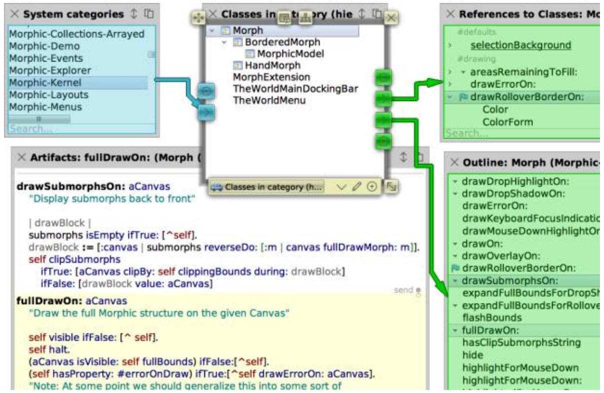
```
" example self: MyMorph new args: {} FormCanvas extent: 60@60   1

aCanvas frameRectangle: self bounds width: 2424@405 color: self color
example  1  > a FormCanvas on: Form(60x60x32)
aCanvas frameOval: self position extent: 30@30 width: 20 color: 🟦
aCanvas
   translateBy: 38.0@33.0
   during: [ :canvas  canvas fillRectangle: self position extent: 20 @ 20 color: 🟩
aCanvas
example  1  > a FormCanvas on: Form(60x60x32)
```



```
MyMorph >> drawOn: aCanvas
" example self: MyMorph new args: {} FormCanvas extent: 60@60   1

aCanvas frameRectangle: self bounds width: 2424@405 color: self color
example  1
aCanvas frameOval: self position extent: 30@30 width: 20 color: 🟦
aCanvas
   translateBy: 38.0@33.0
   during: [ :canvas  canvas fillRectangle: self position extent: 20 @ 20 color: 🟩
aCanvas
example  1
```

Topic: Async Scripts in Vivide

- Extend a tool-building framework to support script interpretation that is not blocking the UI process



```
drawSubmorphsOn: aCanvas
    "Display submorphs back to front"

    | drawBlock |
    submorphs isEmpty ifTrue: [^self].
    drawBlock := [:canvas | submorphs reverseDo: [:m | canvas fullDrawMorph: m]].
    self clipSubmorphs
        ifTrue: [aCanvas clipBy: self clippingBounds during: drawBlock]
        ifFalse: [drawBlock value: aCanvas]
fullDrawOn: aCanvas
    "Draw the full Morphic structure on the given Canvas"

    self visible ifFalse: [^ self].
    self halt.
    (aCanvas isVisible: self fullBounds) ifFalse:[^self].
    (self hasProperty: #errorOnDraw) ifTrue:[^self drawErrorOn: aCanvas].
    "Note: At some point we should generalize this into some sort of
```

- Literature
  - Taeumel, Hirschfeld. Evolving User Interfaces From Within Self-sustaining Programming Environments: Exploring the Project Concept of Squeak/Smalltalk to Bootstrap Uis. In PX/16, 2016, ACM.
  - Taeumel. Data-driven Tool Construction in Exploratory Programming Environments. PhD Thesis, 2020. https://publishup.uni-potsdam.de/44428
    - Pages 10 – 14, 83 – 89, 93 – 103, 131 – 133, 157 – 169

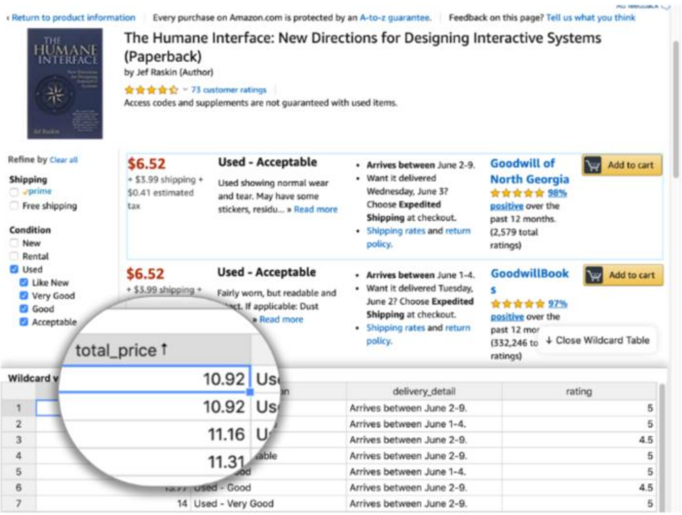
→ github.com/hpi-swa/vivide

Topic: Squeak Inbox Talk

- Improve communication between contributors and core developers by integrating mailing lists and code repositories in a new tool

- Literature (and other Materials)
  - Steinert, Taeumel, Lincke, Pape, Hirschfeld. CodeTalk – Conversations About Code. C5, 2010, IEEE.
  - Ducasse, Nierstrasz, … Squeak By Example (5.3 Edition). 2020, lulu. (Infos about Morphic and tools)
  - github.com/hpi-swa/squeak-history
    - Web scraping mailman in method #downloadTweak
  - github.com/hpi-swa-teaching/IMAPClient
    - Send and receive emails via IMAP

Topic: Wildcard



‹ Return to product information | Every purchase on Amazon.com is protected by an A-to-z guarantee. | Feedback on this page? Tell us what you think

The Humane Interface: New Directions for Designing Interactive Systems (Paperback)

by Jef Raskin (Author)

73 customer ratings

Access codes and supplements are not guaranteed with used items.

Refine by Clear all

Shipping

☐ prime

☐ Free shipping

Condition

☐ New

☐ Rental

☑ Used

☑ Like New

☑ Very Good

☑ Good

☑ Acceptable

$6.52 + $3.99 shipping + $0.41 estimated tax

Used - Acceptable

Used showing normal wear and tear. May have some stickers, residu... » Read more

- **Arrives between** June 2-9.
- Want it delivered Wednesday, June 3? Choose **Expedited Shipping** at checkout.
- Shipping rates and return policy.

Goodwill of North Georgia

98% positive over the past 12 months. (2,579 total ratings)

Add to cart

$6.52 + $3.99 shipping +

Used - Acceptable

Fairly worn, but readable and ...ct. If applicable: Dust ... » Read more

- **Arrives between** June 1-4.
- Want it delivered Tuesday, June 2? Choose **Expedited Shipping** at checkout.
- Shipping rates and return policy.

GoodwillBooks

97% positive over the past 12 mo... (332,246 to... ratings)

Add to cart

↓ Close Wildcard Table

Wildcard v...

total_price ↑

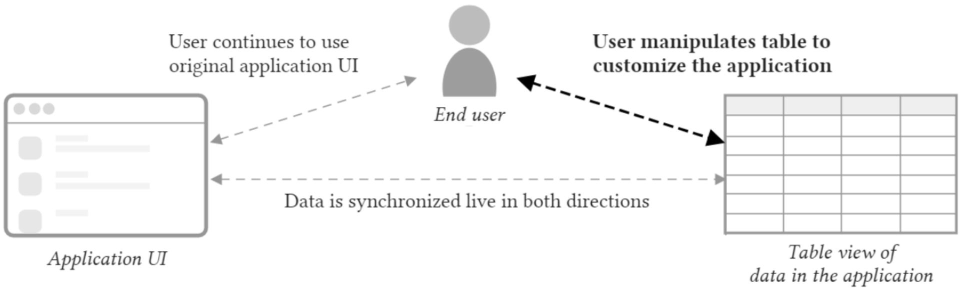
10.92

10.92

11.16

11.31

			delivery_detail	rating
1			Arrives between June 2-9.	5
2			Arrives between June 1-4.	5
3			Arrives between June 2-9.	4.5
4			Arrives between June 2-9.	5
5			Arrives between June 1-4.	5
6		Used - Good	Arrives between June 2-9.	4.5
7	14	Used - Very Good	Arrives between June 2-9.	5



User continues to use original application UI

End user

User manipulates table to customize the application

Data is synchronized live in both directions

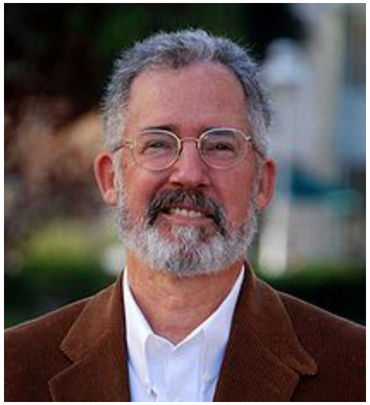
Application UI

Table view of data in the application

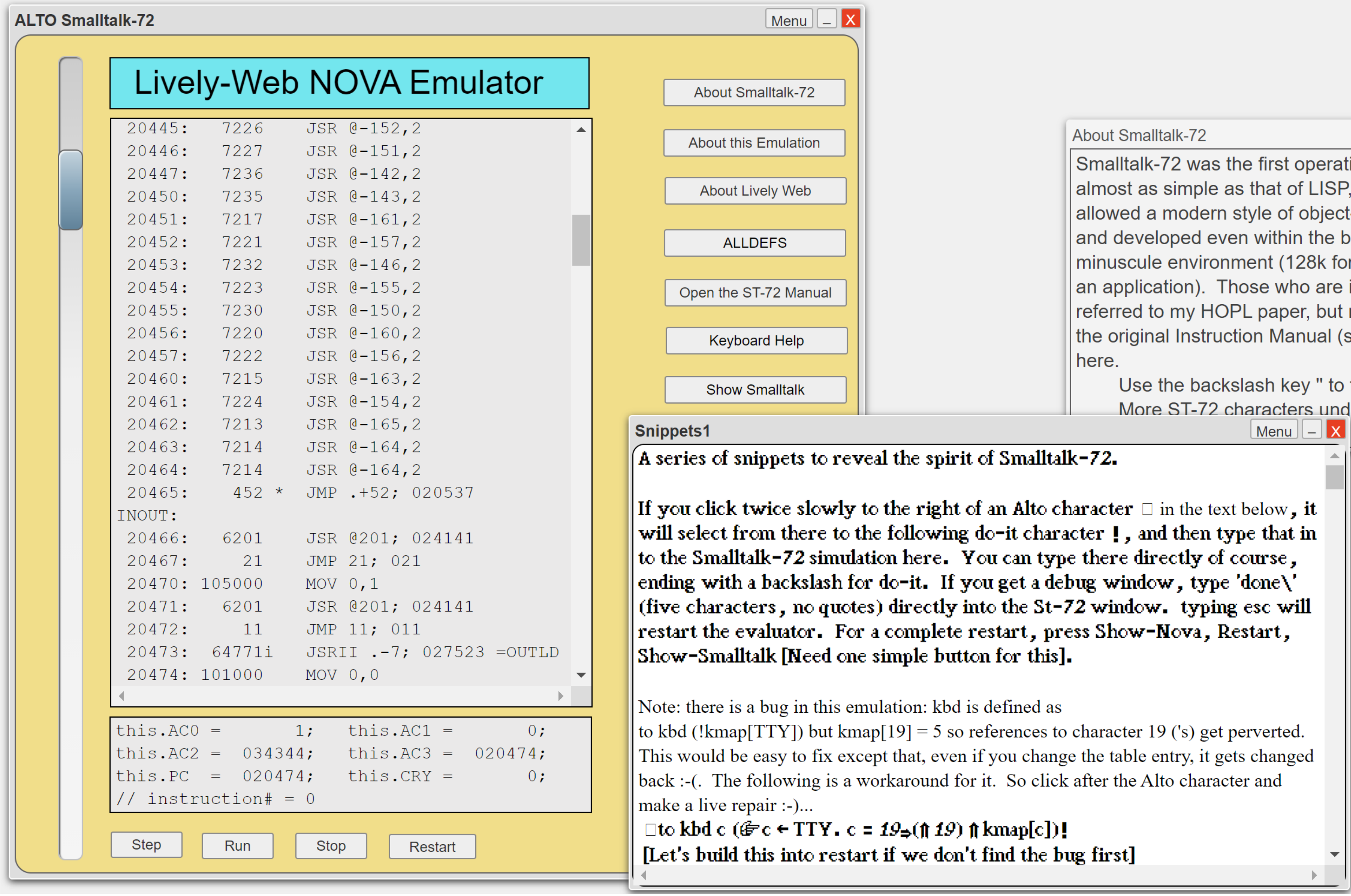
- Literature
  - Geoffrey Litt, Daniel Jackson, Tyler Millis, Jessica Quaye. 2020. End-user software customization by direct manipulation of tabular data.

Topic: Smalltalk-72 redux

- Smalltalk-72 (Alan Kay and Dan Ingalls)
- Goal for Redux
  - metacircular evaluator in St-72 itself
    - currentContext := currentContext next
  - fast enough to be useful /
    faster that the original assembly language interpreter
  - “real definition” of the St-72 interpreter
  - Programmer interface
    - Original had no morphic and no decent debugger



- Literature
  - Daniel Ingalls. 2020. *The evolution of Smalltalk: from Smalltalk-72 through Squeak*
  - https://smalltalkzoo.thechm.org/



```
 20445:   7226    JSR @-152,2
 20446:   7227    JSR @-151,2
 20447:   7236    JSR @-142,2
 20450:   7235    JSR @-143,2
 20451:   7217    JSR @-161,2
 20452:   7221    JSR @-157,2
 20453:   7232    JSR @-146,2
 20454:   7223    JSR @-155,2
 20455:   7230    JSR @-150,2
 20456:   7220    JSR @-160,2
 20457:   7222    JSR @-156,2
 20460:   7215    JSR @-163,2
 20461:   7224    JSR @-154,2
 20462:   7213    JSR @-165,2
 20463:   7214    JSR @-164,2
 20464:   7214    JSR @-164,2
 20465:    452 *  JMP .+52; 020537
INOUT:
 20466:   6201    JSR @201; 024141
 20467:     21    JMP 21; 021
 20470: 105000    MOV 0,1
 20471:   6201    JSR @201; 024141
 20472:     11    JMP 11; 011
 20473:  64771i   JSRII .-7; 027523 =OUTLD
 20474: 101000    MOV 0,0
```

```
this.AC0 =       1;   this.AC1 =       0;
this.AC2 =  034344;   this.AC3 =  020474;
this.PC  =  020474;   this.CRY =       0;
// instruction# = 0
```

Smalltalk-72 was the first operati
almost as simple as that of LISP,
allowed a modern style of object
and developed even within the b
minuscule environment (128k for
an application). Those who are i
referred to my HOPL paper, but r
the original Instruction Manual (s
here.

Use the backslash key " to
More ST-72 characters und

A series of snippets to reveal the spirit of Smalltalk-72.

If you click twice slowly to the right of an Alto character □ in the text below, it will select from there to the following do-it character !, and then type that in to the Smalltalk-72 simulation here. You can type there directly of course, ending with a backslash for do-it. If you get a debug window, type 'done\' (five characters, no quotes) directly into the St-72 window. typing esc will restart the evaluator. For a complete restart, press Show-Nova, Restart, Show-Smalltalk [Need one simple button for this].

Note: there is a bug in this emulation: kbd is defined as to kbd (!kmap[TTY]) but kmap[19] = 5 so references to character 19 ('s) get perverted. This would be easy to fix except that, even if you change the table entry, it gets changed back :-(. The following is a workaround for it. So click after the Alto character and make a live repair :-)...

□to kbd c (☞c ← TTY. c = 19⇒(⇑ 19) ⇑kmap[c])!
[Let's build this into restart if we don't find the bug first]

Bring Your Own Project Idea

- Project proposals are inspirations
- You can shape direction of project!
  - *which use case to cover, …*

Development Links

- Squeak/Smalltalk
  - https://squeak.org/
  - https://github.com/hpi-swa/vivide
- Lively4:
  - https://github.com/livelykernel/lively4-core
  - https://lively-kernel.org/lively4/lively4-core/start.html

Hand-In Reference Sheet

- Presentation
  - Presentation (pdf)
  - Screencast (mp4, public, 1080p)
- End of Semester (in zip, link or email)
  - Presentation and Screencast again
  - Abstract (txt)
  - Figure / Screenshot (png, 800x600)
  - Sourcecode (MIT License)
  - Data
  - (squeak image)
  - README.md
- All hand-ins will be archived in a public repository

Organization

- Course
  - Project-Seminar, 4 SWS, 1-2 students per group
- Presentations
  - End-term Presentation (July)
- Grading
  - 6 ECTS graded credit points
  - Grade based on project work and presentation
- Important dates
  - Project topics on April 15th
  - Enrollment with preferred topic names on or before April 20th
    - Mail to jens.lincke@hpi.de with **Live21** in subject
  - Topic assignment on April 22th
  - Presentation dates determined after topics are assigned