

The Moldable Inspector

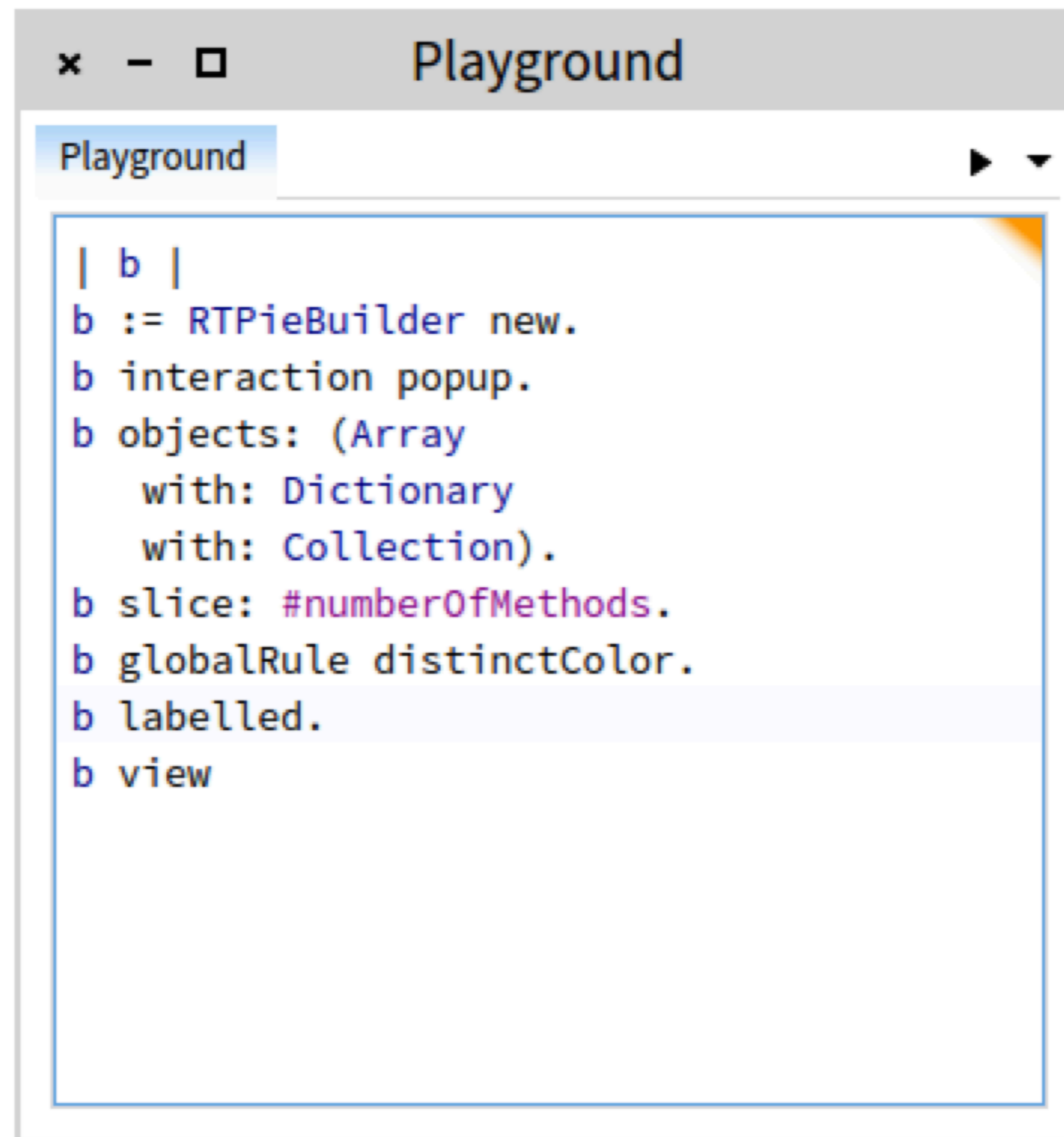
A Framework for Domain-Specific Object Inspection

Andrei Chiş
University of Bern

Tudor Girba
CompuGroup Medical
Schweiz AG

Oscar Nierstrasz
University of Bern

Finding a conflicting keybinding in a text morph



```
| b |  
b := RTPieBuilder new.  
b interaction popup.  
b objects: (Array  
  with: Dictionary  
  with: Collection).  
b slice: #numberOfMethods.  
b globalRule distinctColor.  
b labelled.  
b view
```

Variable	Value
self	a RubScrolledTextMorph(620494848)
alwaysAccept	nil
autoAccept	false
bounds	(6.0@30.0) corner: (341.0@331.0)
corner	(341.0@331.0)
origin	(6.0@30.0)
color	Color white
alpha	255
cachedBitPattern	a Bitmap of length 1
cachedDepth	32
rgb	1073741823
extension	a MorphExtension (661651...her -> a KMDispatcher)]
fullBounds	(6@30) corner: (341@331)
getSelectionSelector	nil
getTextSelector	#getText
hasEditingConflicts	false

```
"a RubScrolledTextMorph(620494848) "
self
```

The image displays five overlapping Inspector windows in a software development environment, each showing the state and variables of a specific object:

- Inspector on a RubScrolledTextMorph (a RubScrolledTextMorph(620494848))**: Shows a variable `self`.
- Inspector on a KMDispatcher (a KMDispatcher)**: Shows a variable `self`.
- Inspector on a KMCategory (RubTextEditor)**: Shows a variable `self`.
- Inspector on a Set (a Set(#cutCommand on Cmd + X do [:t...t] target editor undo)**: Shows a list of 23 items, including:
 - `#cutCommand on Cmd + X do [:t...t] target editor undo`
 - `#copySelectionCommand on Cmd + X do [:t...t] target editor undo`
 - `#outdentCommand on Shift + Ctrl + X do [:t...t] target editor undo`
 - `#undoCommand on Cmd + Z do [:target | target editor undo]` (highlighted)
 - `#cancelCommand on Cmd + X do [:t...t] target editor undo`
 - `#selectAllCommand on Cmd + A do [:t...t] target editor undo`
 - `#acceptCommand on Cmd + Enter do [:t...t] target editor undo`
 - `#findAgain on Ctrl + G do [:t...t] target editor undo`
 - `#paste on Ctrl + V do [:t...t] target editor undo`
 - `#copySelection on Ctrl + C do [:t...t] target editor undo`
 - `#undo on Ctrl + Z do [:target | target editor undo]`
 - `#outdent on Shift + Ctrl + X do [:t...t] target editor undo`
 - `#cut on Ctrl + X do [:t...t] target editor undo`
- Inspector on a KMKeymap (#undoCommand on Cmd + Z ... | target editor undo)**: Shows a table of variables and their values:

Variable	Value
<code>self</code>	<code>#undoCommand on Cmd + Z ... target editor undo</code>
<code>action</code>	<code>[:target target editor undo]</code>
<code>defaultShortcut</code>	<code>Cmd + Z</code>
<code>description</code>	<code>'Undo'</code>
<code>name</code>	<code>#undoCommand</code>
<code>shortcut</code>	<code>Cmd + Z</code>

Different tasks exercise
different aspects of an object

Different tasks exercise
different aspects of an object

We rarely look at a
single object in isolation

Moldable presentations

multiple interchangeable
presentations for each object

What about the keybindings?

The screenshot shows the Inspector window for a RubEditingArea object. The 'Keys' tab is selected, displaying a table of keybindings. The table has four columns: Shortcut, Description, Action, and Category. The shortcuts listed include Shift + Ctrl + Z (Redo), Shift + Cmd + R (Indent), Ctrl + C (Copy selection), Shift + Cmd + E (Method strings containing it), Cmd + A (Select all), Ctrl + B (Browse), Cmd + Z (Undo), Shift + Ctrl + D (Debug it), Cmd + V (Paste), Cmd + O (Paste), Cmd + N (Senders of it), Cmd + L (Cancel unsaved editings), Ctrl + P (Paste), Ctrl + V (Paste), and Shift + Cmd + D (Debug it). The categories are either RubTextEditor or RubSmalltalkEditor. A search bar at the bottom contains the text 'enter search query'.

Shortcut	Description	Action	Category
Shift + Ctrl + Z	Redo	[:target target editor redo]	RubTextEditor
Shift + Cmd + R	Indent	[:target target editor indent: nil]	RubTextEditor
Ctrl + C	Copy selection	[:target target editor copySelection	RubTextEditor
Shift + Cmd + E	Method strings containing it	[:target target editor methodString	RubSmalltalkEditor
Cmd + A	Select all	[:target target editor selectAll: nil]	RubTextEditor
Ctrl + B	Browse	[:target target editor browselt: nil]	RubSmalltalkEditor
Cmd + Z	Undo	[:target target editor undo]	RubTextEditor
Shift + Ctrl + D	Debug it	[:target target editor debugIt: nil]	RubSmalltalkEditor
Cmd + V	Paste	[:target target editor paste]	RubTextEditor
Cmd + O	Paste	[self actOn: aPresentation]	
Cmd + N	Senders of it	[:target target editor sendersOfIt: r	RubSmalltalkEditor
Cmd + L	Cancel unsaved editings	[:target target editor cancel]	RubTextEditor
Ctrl + P	Paste	[self actOn: aPresentation]	
Ctrl + V	Paste	[:target target editor paste]	RubTextEditor
Shift + Cmd + D	Debug it	[:target target editor debugIt: nil]	RubSmalltalkEditor

What about the source code?

Inspector on a RubEditingArea (a RubEditingArea(408158208))

a RubEditingArea (a RubEditingArea(408158208))

State Extension Submorphs Morph Keys Meta

RubEditingArea
RubAbstractTextArea
Morph
Object
ProtoObject

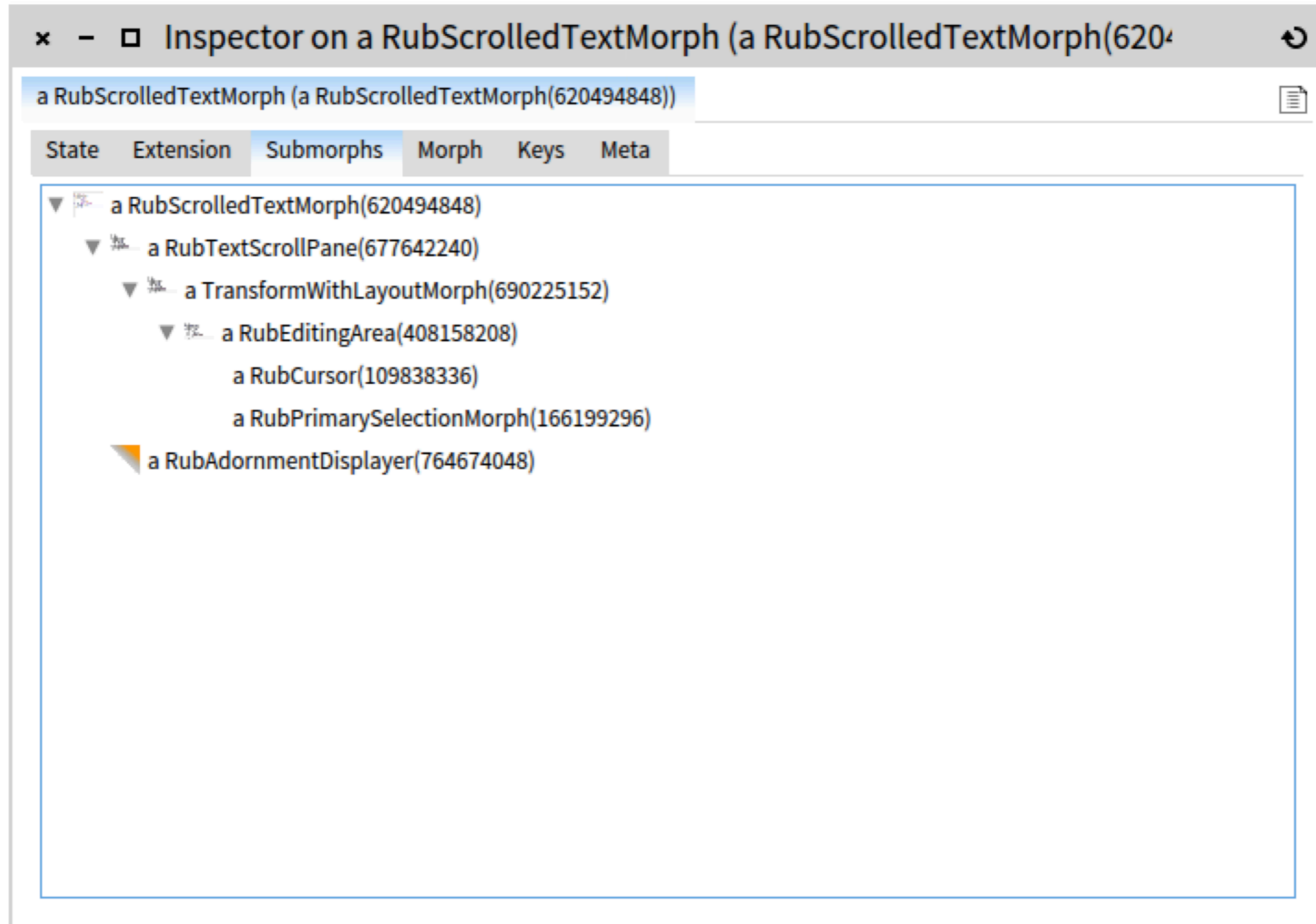
hideOverEditableTextCursor event handling
initialize initialize
initializeShortcuts: keymapping
keyStroke: event handling
keyboardFocusChange: event handling
lineNumbersBackgroundColor accessing
lineNumbersTextColor accessing

enter search query

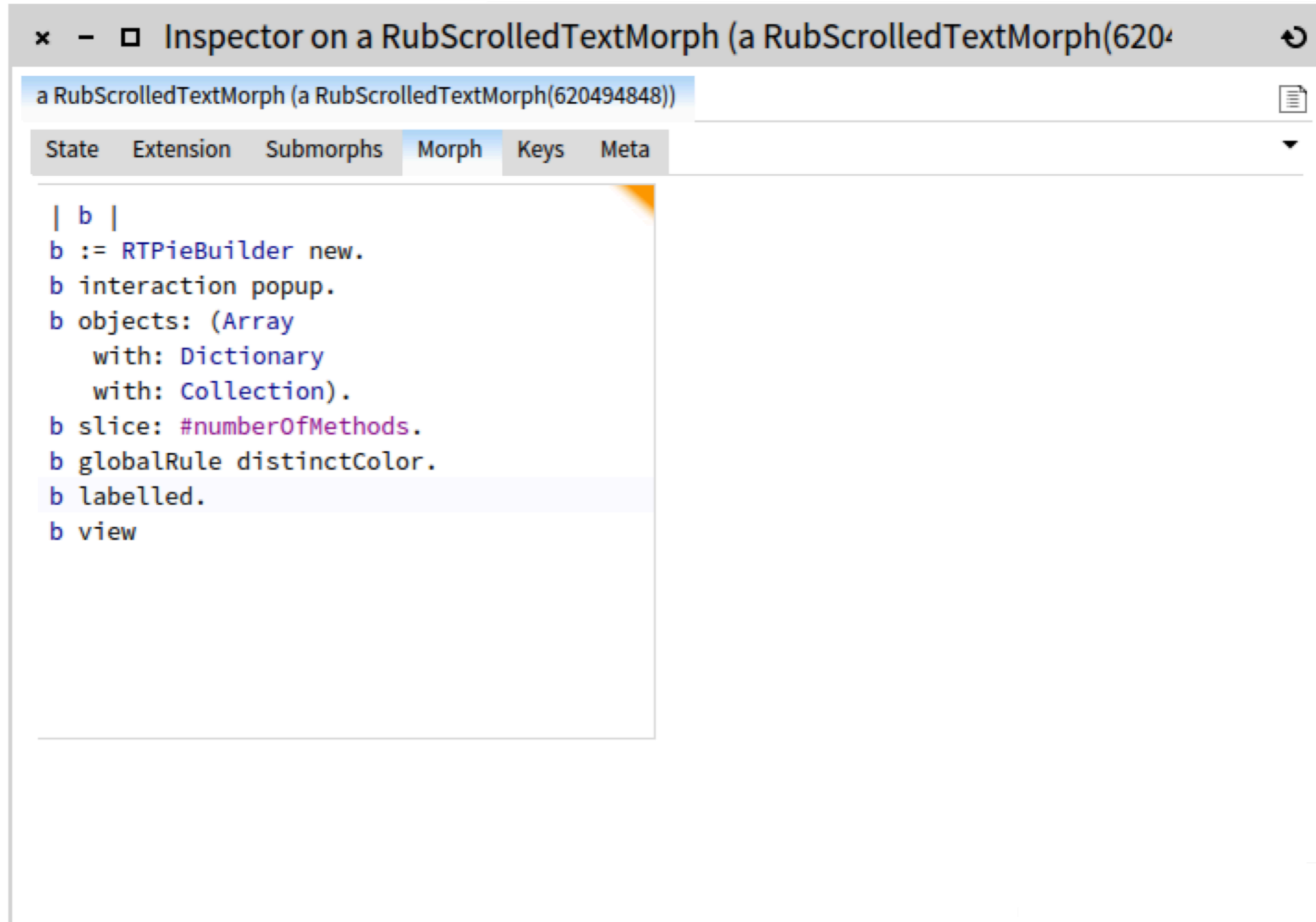
✓ i

```
initialize  
  super initialize.  
  self compose.  
  self addCursor.  
  self plugFindReplace.
```

What about the submorphs?



What about the morph?



The screenshot shows a Morph Inspector window titled "Inspector on a RubScrolledTextMorph (a RubScrolledTextMorph(620494848))". The "Morph" tab is selected, displaying the following code:

```
| b |  
b := RTPieBuilder new.  
b interaction popup.  
b objects: (Array  
  with: Dictionary  
  with: Collection).  
b slice: #numberOfMethods.  
b globalRule distinctColor.  
b labelled.  
b view
```

Moldable presentations

Inspector on a RubScrolledTextMorph (a RubScrolledTextMorph(620494848))

a RubScrolledTextMorph (a RubScrolledTextMorph(620494848))

Variable	Value
self	a RubScrolledTextMorph(620494848)
alwaysAccept	nil
autoAccept	false
bounds	(6.0@30.0) corner: (341.0@331.0)
color	Color white
extension	a MorphExtension (661651...her -> a KMDispatcher)
fullBounds	(6@30) corner: (341@331)
getSelectionSelector	nil

```
"a RubScrolledTextMorph(620494848)"
self textArea
```

Inspector on a RubScrolledTextMorph (a RubScrolledTextMorph(620494848))

a RubScrolledTextMorph (a RubScrolledTextMorph(620494848))

Variable	Value
actionMap	nil
balloonText	nil
borderStyle	(SimpleBorder width: 1 c...2 b: 0.862 alpha: 1.0)
clipSubmorphs	true
cornerStyle	nil
eventHandler	nil
externalName	nil
fillStyle	nil
layoutFrame	nil
layoutPolicy	nil
layoutProperties	a LayoutProperties
locked	false
otherProperties	a SmallIdentityDictionary...atcher->a KMDispatcher)
sticky	false
visible	true

Inspector on a RubEditingArea (a RubEditingArea(408158208))

a RubEditingArea (a RubEditingArea(408158208))

Shortcut	Description	Action	Category
Shift + Ctrl + Z	Redo	[:target target editor redo]	RubTextEditor
Shift + Cmd + R	Indent	[:target target editor indent: nil]	RubTextEditor
Ctrl + C	Copy selection	[:target target editor copySelection]	RubTextEditor
Shift + Cmd + E	Method strings containing it	[:target target editor methodString]	RubSmalltalkEditor
Cmd + A	Select all	[:target target editor selectAll: nil]	RubTextEditor
Ctrl + B	Browse	[:target target editor browse: nil]	RubSmalltalkEditor
Cmd + Z	Undo	[:target target editor undo]	RubTextEditor
Shift + Ctrl + D	Debug it	[:target target editor debugit: nil]	RubSmalltalkEditor
Cmd + V	Paste	[:target target editor paste]	RubTextEditor
Cmd + O		[self actOn: aPresentation]	
Cmd + N	Senders of it	[:target target editor sendersOfit: nil]	RubSmalltalkEditor
Cmd + L	Cancel unsaved editings	[:target target editor cancel]	RubTextEditor
Ctrl + P		[self actOn: aPresentation]	
Ctrl + V	Paste	[:target target editor paste]	RubTextEditor
Shift + Cmd + D	Debug it	[:target target editor debugit: nil]	RubSmalltalkEditor

enter search query

Inspector on a RubEditingArea (a RubEditingArea(408158208))

a RubEditingArea (a RubEditingArea(408158208))

Variable	Value
RubEditingArea	hideOverEditableTextCursor event handling
RubAbstractTextArea	initialize initialize
Morph	initializeShortcuts: keymapping
Object	keyStroke: event handling
ProtoObject	keyboardFocusChange: event handling
	lineNumbersBackgroundColor: accessing
	lineNumbersTextColor: accessing

enter search query

```
initialize
super initialize.
self compose.
self addCursor.
self plugFindReplace.
```

Inspector on a RubScrolledTextMorph (a RubScrolledTextMorph(620494848))

a RubScrolledTextMorph (a RubScrolledTextMorph(620494848))

- a RubTextScrollPane(677642240)
 - a TransformWithLayoutMorph(690225152)
 - a RubEditingArea(408158208)
 - a RubCursor(109838336)
 - a RubPrimarySelectionMorph(166199296)
 - a RubAdornmentDisplayer(764674048)

Inspector on a RubScrolledTextMorph (a RubScrolledTextMorph(620494848))

a RubScrolledTextMorph (a RubScrolledTextMorph(620494848))

```
| b |
b := RTPieBuilder new.
b interaction popup.
b objects: (Array
with: Dictionary
with: Collection).
b slice: #numberOfMethods.
b globalRule distinctColor.
b labelled.
b view
```

Moldable navigation

a workflow in which multiple levels of connecting objects can be seen together

The image displays five overlapping Inspector windows in a software development environment. The windows are as follows:

- Inspector on a RubScrolledTextMorph (a RubScrolledTextMorph(620494848))**: Shows a variable named `self`.
- Inspector on a KMDispatcher (a KMDispatcher)**: Shows a variable named `self`.
- Inspector on a KMCategory (RubTextEditor)**: Shows a variable named `self`.
- Inspector on a Set (a Set(#cutCommand on Cmd + X do [:target | target editor undo]))**: Shows a list of 23 items, including `#undoCommand on Cmd + Z do [:target | target editor undo]`.
- Inspector on a KMKeymap (#undoCommand on Cmd + Z ... | target editor undo)**: Shows a table of variables and their values.

Variable	Value
<code>self</code>	<code>#undoCommand on Cmd + Z ... target editor undo</code>
<code>action</code>	<code>[:target target editor undo]</code>
<code>defaultShortcut</code>	<code>Cmd + Z</code>
<code>description</code>	<code>'Undo'</code>
<code>name</code>	<code>#undoCommand</code>
<code>shortcut</code>	<code>Cmd + Z</code>

Inspector on a KMDispatcher

a KMDispatcher (a KMDispatcher)

State Keys Meta

Variable	Value
self	a KMDispatcher
currentEvent	nil
directKeymaps	nil
morph	a RubEditingArea(408158208)
target	a KMTarget
targets	a Set(aKMCategoryTarget(...phFocusCtrlN:

```
"a KMDispatcher"
self targets asOrderedCollection first
category
```

Inspector on a KMCategory (I

a KMCategory (RubTextEditor)

State Keys Meta

Variable	Value
self	RubTextEditor
name	#RubTextEditor
platforms	a Dictionary(#MacOSX->a ...age #all->a KMS

```
"a KMCategory"
self allEntries keymaps|
```

Inspector on a Set (a Set(#cu

a Set (a Set(#cutCommand on Cmd... editor find: nil] ...)

State 23 items Meta

- #cutCommand on Cmd + X d...t | target editor cut]
- #copySelectionCommand on... editor copySelection]
- #outdentCommand on Shift...t editor outdent: nil]
- #undoCommand on Cmd + Z ... | target editor undo]
- #cancelCommand on Cmd + ... target editor cancel]
- #selectAllCommand on Cmd...editor selectAll: nil]
- #acceptCommand on Cmd + ... target editor accept]
- #findAgain on Ctrl + G d...editor findAgain: nil]
- #paste on Ctrl + V do [...] target editor paste]
- #copySelection on Ctrl +... editor copySelection]
- #undo on Ctrl + Z do [:... | target editor undo]
- #outdent on Shift + Ctrl...t editor outdent: nil]
- #cut on Ctrl + X do [:t...t | target editor cut]
- #selectAll on Ctrl + A d...editor selectAll: nil]
- #findAgain on Ctrl + G d...editor findAgain: nil]

enter search query

Inspector on a KMKeymap (#

a KMKeymap (#undoCommand on Cmd + Z ... | target editor undo])

State Source code Meta

Variable	Value
self	#undoCommand on Cmd + Z ... target ed
action	[:target target editor undo]
defaultShortcut	Cmd + Z
description	'Undo'
name	#undoCommand
shortcut	Cmd + Z

```
"#undoCommand on Cmd + Z do [ :target | target
editor undo ]
"
self
```

Inspector on a RubScrolledTextMorph (a RubScrolledTextMorph(620494848))

a KMDispatcher (a KMDispatcher)

State Keys Meta

Variable	Value
self	a KMDispatcher
currentEvent	nil
directKeymaps	nil
morph	a RubEditingArea(408
target	a KMTarget
targets	a Set(aKMCategoryTa

```
"a KMDispatcher"
self targets
asOrderedCollection first
category
```

a KMCategory (RubTextEditor)

State Keys Meta

Variable	Value
self	RubTextEditor
name	#RubTextEditor
platforms	a Dictionary(#MacOSX->a .

```
"a KMCategory"
self allEntries keymaps
```

a Set (a Set(#cutCommand on Cmd... editor find

State 23 items Meta

```
#cutCommand on Cmd + X d...t | target editor c
#copySelectionCommand on... editor copySele
#outdentCommand on Shift...t editor outdent:
#undoCommand on Cmd + Z ... | target editor u
#cancelCommand on Cmd + ... target editor ca
#selectAllCommand on Cmd...editor selectAll: r
#acceptCommand on Cmd + ... target editor ac
#findAgain on Ctrl + G d...editor findAgain: nil ]
#paste on Ctrl + V do [ ...] target editor paste ]
#copySelection on Ctrl +... editor copySelector
#undo on Ctrl + Z do [ :... | target editor undo ]
#outdent on Shift + Ctrl...t editor outdent: nil ]
#cut on Ctrl + X do [ :t...t | target editor cut ]
#selectAll on Ctrl + A d...editor selectAll: nil ]
#find on Cmd + F do [ :t...t | target editor find: nil ]
enter search query
```

a KMKeymap (#undo on Ctrl + Z do [:... | target e

State Source code Meta

Variable	Value
self	#undo on Ctrl + Z do [:...
action	[:target target editor
defaultShortcut	Ctrl + Z
description	'Undo'
name	#undo
shortcut	Ctrl + Z

```
"#undo on Ctrl + Z do [ :target
undo ]|"
self
```


Inspector on a RubScrolledTextMorph (a RubScrolledTextMorph(620494848))

a Set (a Set(#cutCommand on Cmd... editor find: nil] ...)

State 23 items Meta

- #cutCommand on Cmd + X d...t | target editor cut]
- #copySelectionCommand on... editor copySelection]
- #outdentCommand on Shift...t editor outdent: nil]
- #undoCommand on Cmd + Z ... | target editor undo]
- #cancelCommand on Cmd + ... target editor cancel]
- #selectAllCommand on Cmd...editor selectAll: nil]
- #acceptCommand on Cmd + ... target editor accept]
- #findAgain on Ctrl + G d...editor findAgain: nil]
- #paste on Ctrl + V do [...] target editor paste]
- #copySelection on Ctrl +... editor copySelection]
- #undo on Ctrl + Z do [:... | target editor undo]
- #outdent on Shift + Ctrl...t editor outdent: nil]
- #cut on Ctrl + X do [:t...t | target editor cut]
- #selectAll on Ctrl + A d...editor selectAll: nil]
- #find on Cmd + F do [:t...t | target editor find: nil]

enter search query

a KMKeymap (#undo on Ctrl + Z do [:... | target editor undo])

State Source code Meta

Variable	Value
self	#undo on Ctrl + Z do [:... target editor undo]
action	[:target target editor undo]
defaultShortcut	Ctrl + Z
description	'Undo'
name	#undo
shortcut	Ctrl + Z

```
"#undo on Ctrl + Z do [ :target | target editor
undo ]"
self
```

Inspector on a RubScrolledTextMorph (a RubScrolledTextMorph(620494848))

a Set (a Set(#cutCommand on Cmd... editor find: nil] ...)

State 23 items Meta

- #cutCommand on Cmd + X d...t | target editor cut]
- #copySelectionCommand on... editor copySelection]
- #outdentCommand on Shift...t editor outdent: nil]
- #undoCommand on Cmd + Z ... | target editor undo]
- #cancelCommand on Cmd + ... target editor cancel]
- #selectAllCommand on Cmd...editor selectAll: nil]
- #acceptCommand on Cmd + ... target editor accept]
- #findAgain on Ctrl + G d...editor findAgain: nil]
- #paste on Ctrl + V do [...| target editor paste]
- #copySelection on Ctrl +... editor copySelection]
- #undo on Ctrl + Z do [:... | target editor undo]
- #outdent on Shift + Ctrl...t editor outdent: nil]
- #cut on Ctrl + X do [:t...t | target editor cut]
- #selectAll on Ctrl + A d...editor selectAll: nil]
- #find on Cmd + F do [t...t | target editor find: nil]

enter search query

a KMKeymap (#undo on Ctrl + Z do [:... | target editor undo])

State Source code Meta

Variable	Value
self	#undo on Ctrl + Z do [:... target editor undo]
action	[:target target editor undo]
defaultShortcut	Ctrl + Z
description	'Undo'
name	#undo
shortcut	Ctrl + Z

```
"#undo on Ctrl + Z do [ :target | target editor
undo ]"
self
```

The Moldable Inspector

Moldable presentations + Moldable navigation

Browse keybindings

The screenshot shows the Inspector window for a RubEditingArea object. The left pane displays a table of keybindings, and the right pane shows the source code for a KMKeymap object.

Shortcut	Description	Action
Shift + Ctrl + Z	Redo	[:target target editor redo]
Cmd + L	Cancel unsaved editings	[:target target editor cancel]
Shift + Ctrl + L	Outdent	[:target target editor outdent: nil]
Ctrl + O		[self actOn: aPresentation]
Cmd + Z	Undo	[:target target editor undo]
Shift + Cmd + L	Outdent	[:target target editor outdent: nil]
Shift + Ctrl + TAB		[:target :morph :event morph navi
Ctrl + G	Find text again	[:target target editor findAgain: ni
Shift + Cmd + R	Indent	[:target target editor indent: nil]
Cmd + S	Accept unsaved editings	[:target target editor accept]
Cmd + X	Cut selection	[:target target editor cut]
Ctrl + C	Copy selection	[:target target editor copySelectio
Ctrl + TAB		[:target :morph :event morph navi
Cmd + O		[self actOn: aPresentation]
Ctrl + Z	Undo	[:target target editor undo]

```
(aBuilder shortcut: #undoCommand)
  category: RubTextEditor name
  default: $Z command
  do: [ :target | target editor undo ]
  description: 'Undo'.

(aBuilder shortcut: #redo)
  category: RubTextEditor name
  default: $Z shift ctrl
  do: [ :target | target editor redo ]
  description: 'Redo'.

(aBuilder shortcut: #redoCommand)
  category: RubTextEditor name
  default: $Z shift command
  do: [ :target | target editor redo ]
  description: 'Redo'.

(aBuilder shortcut: #find)
  category: RubTextEditor name
  default: $f ctrl
```

Browse AST nodes

The image shows a screenshot of the Ruby Inspector application. The title bar reads "Inspector on a CompiledMethod (Object>>#printStringLimitedTo:)".

The left pane shows the AST for a CompiledMethod (Object>>#printStringLimitedTo:). The tree structure is as follows:

- ▼ RBMethodNode(printStringLimitedTo: limit "Answer a String whos)
 - RBArgumentNode(limit)
 - ▼ RBSequenceNode(| limitedString | limitedString := String streamCo)
 - RBTemporaryNode(limitedString)
 - ▶ RBAssignmentNode(limitedString := String streamContents: [:s | sel)
 - ▶ RBMessageNode(limitedString size < limit ifTrue: [^ limitedStr)
 - ▶ RBReturnNode(^ limitedString, '...etc...')

a RBAssignmentNode (RBAssignmentNode(limited...: s] limitedTo: limit))

State Source code Tree Meta

```
printStringLimitedTo: limit  
"Answer a String whose characters are a description of  
the receiver.  
If you want to print without a character limit, use  
fullPrintString."  
| limitedString |  
limitedString := String streamContents: [:s |  
self printOn: s] limitedTo: limit.  
limitedString size < limit ifTrue: [^ limitedString].  
^ limitedString , '...etc...'
```

Explore files

Inspector on a FileReference (File @ /Users/andrei/PHD...-innovation-award/image)

a FileReference (File @ /Users/andrei/PHD...-innovation-award/image)


Name	Size	Creation
..	136 B	2014-08-10 11:28:47
data	204 B	2014-07-27 19:21:00
package-cache	306 B	2014-08-10 11:31:45
pharo-vm	238 B	2014-08-10 11:29:38
.DS_Store	6.15 kB	2014-08-10 12:18:49
Moose-iwst-screenshots.changes	17.85 MB	2014-08-12 00:45:21
Moose-iwst-screenshots.image	50.90 MB	2014-08-12 00:45:21
Moose.changes	17.85 MB	2014-08-10 02:25:47
Moose.image	47.27 MB	2014-08-10 02:24:35
pharo	382 B	2014-08-10 11:29:42
pharo-ui	371 B	2014-08-10 11:29:42
PharoDebug.log	188.73 kB	2014-08-10 11:31:41

a FileReference (File @ /Users/andrei/PHD...-innovation-award/image/data)

Name	Size
..	442 B
scripts	136 B
.DS_Store	6.15 kB
flower.png	11.09 kB
README.txt	390 B
script.st	271 B

a FileReference (File @ /Users/andrei/PHD...-innovation-award/image/data/flower.png)

State Raw Picture Meta



Explore files

The screenshot shows a file explorer interface with three panels. The top panel is titled "Inspector on a FileReference (File @ /Users/andrei/PHD...-innovation-award/image)".

The left panel shows a file list with columns "Name", "Size", and "Creation". The "data" folder is selected.

Name	Size	Creation
..	136 B	2014-08-10 11:28:47
data	204 B	2014-07-27 19:21:00
package-cache	306 B	2014-08-10 11:31:45
pharo-vm	238 B	2014-08-10 11:29:38
.DS_Store	6.15 kB	2014-08-10 12:18:49
Moose-iwst-screenshots.changes	17.85 MB	2014-08-12 00:45:21
Moose-iwst-screenshots.image	50.90 MB	2014-08-12 00:45:21
Moose.changes	17.85 MB	2014-08-10 02:25:47
Moose.image	47.27 MB	2014-08-10 02:24:35
pharo	382 B	2014-08-10 11:29:42
pharo-ui	371 B	2014-08-10 11:29:42
PharoDebug.log	188.73 kB	2014-08-10 11:31:41

The middle panel shows a sub-directory list with columns "Name" and "Size". The "script.st" file is selected.

Name	Size
..	442 B
scripts	136 B
.DS_Store	6.15 kB
flower.png	11.09 kB
README.txt	390 B
script.st	271 B

The right panel shows the content of the selected "script.st" file, which is a Smalltalk script:

```
| b |  
b := RTPieBuilder new.  
b interaction popup.  
b objects: (Array with: Dictionary with: Collection  
with: Set with: OrderedCollection ).  
b slice: #numberOfMethods.  
b globalRule distinctColor.  
b labelled.  
b emphasize: Set by: 20.  
b emphasize: Dictionary by: 20.  
b view
```

The Moldable Inspector

task-specific workflows
directly within the inspector

gt.moosetechnology.org

moose