

Visualization of Object Sets in Pharo

Presentation BA Thesis
Software Composition Group

Eve Mendoza Quiros
Supervisor: Claudio Corrodi

March 4th 2018

Motivation

Currently

- No overview is conveyed
- Relationships between objects are not shown clearly

The screenshot shows a debugger window titled "a LinkedList [6 items] (Color yellow Color blue Color red Color green (Color r: 0.867 g: 0.45... x". The window has tabs for "Items", "Raw", and "Meta". The "Raw" tab is selected, displaying a tree view of the object's structure. The tree starts with a root object containing a "self" property. It then branches into "firstLink", "nextLink", and "lastLink". Each link contains a "self" property and a "nextLink" property. The "nextLink" property eventually points to a "nil" object. The "value" property of each link is represented by a colored square: yellow, blue, red, green, and yellow. The "value" property of the final link is represented by a pink square. The "value" property of the "nil" object is represented by a dark green square.

Variable	Value
{ } self	a LinkedList [6 items] (Color yellow Color blue Color red Color green (Color r: 0.867 g: 0.45... x
▼ firstLink	ValueLink(Color yellow)
self	ValueLink(Color yellow)
▼ nextLink	ValueLink(Color blue)
self	ValueLink(Color blue)
▼ nextLink	ValueLink(Color red)
self	ValueLink(Color red)
▼ nextLink	ValueLink(Color green)
self	ValueLink(Color green)
▼ nextLink	ValueLink((Color r: 0.867 g: 0.436 b: 0.495 alpha: 1.0))
self	ValueLink((Color r: 0.867 g: 0.436 b: 0.495 alpha: 1.0))
▼ nextLi	ValueLink((Color r: 0.194 g: 0.443 b: 0.458 alpha: 1.0))
self	ValueLink((Color r: 0.194 g: 0.443 b: 0.458 alpha: 1.0))
▼ n nil	nil
value	(Color r: 0.194 g: 0.443 b: 0.458 alpha: 1.0)
value	(Color r: 0.867 g: 0.436 b: 0.495 alpha: 1.0)
value	Color green
value	Color red
value	Color blue
value	Color yellow
lastLink	ValueLink((Color r: 0.194 g: 0.443 b: 0.458 alpha: 1.0))

Motivation

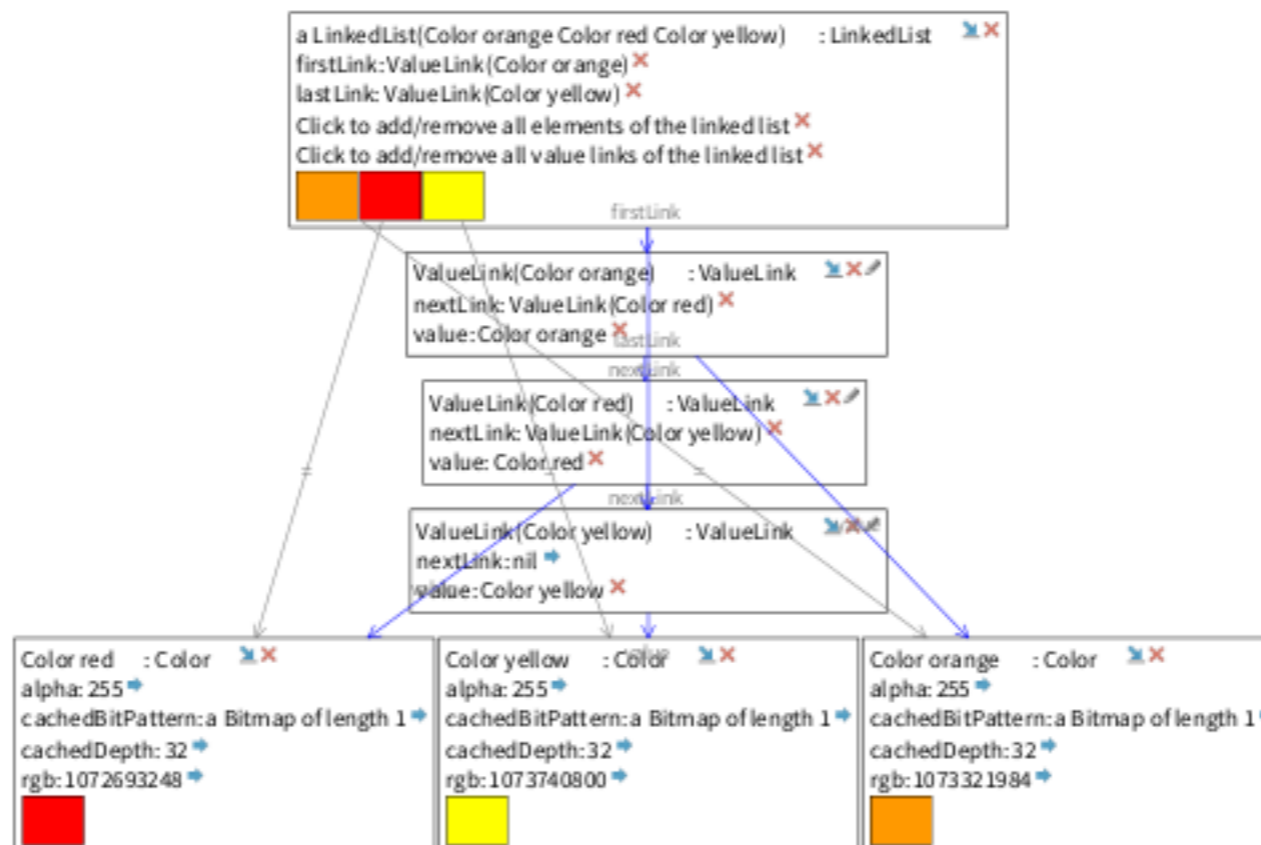
Goal

- Ease code inspection and debugging
- Inspection of multiple objects at the same time

The screenshot displays a 'Playground' window with two object inspection panels. The left panel, titled 'a LargePositiveInteger (4294901760)', shows a table with columns 'Variable' and 'Value'. The table contains five rows: 'self' (4294901760), '1' (0), '2' (0), '3' (255), and '4' (255). The '2' row is highlighted. Below the table, the text '"4294901760"' and 'self' is displayed. The right panel, titled 'a SmallInteger (0)', shows a table with columns 'Variable' and 'Value'. The table contains one row: 'self' (0). Below the table, the text '"0"' and 'self' is displayed. At the bottom center of the interface, a row of six small circles is highlighted with a red box.

Our approach

Visualize object sets and their relationships as a graph

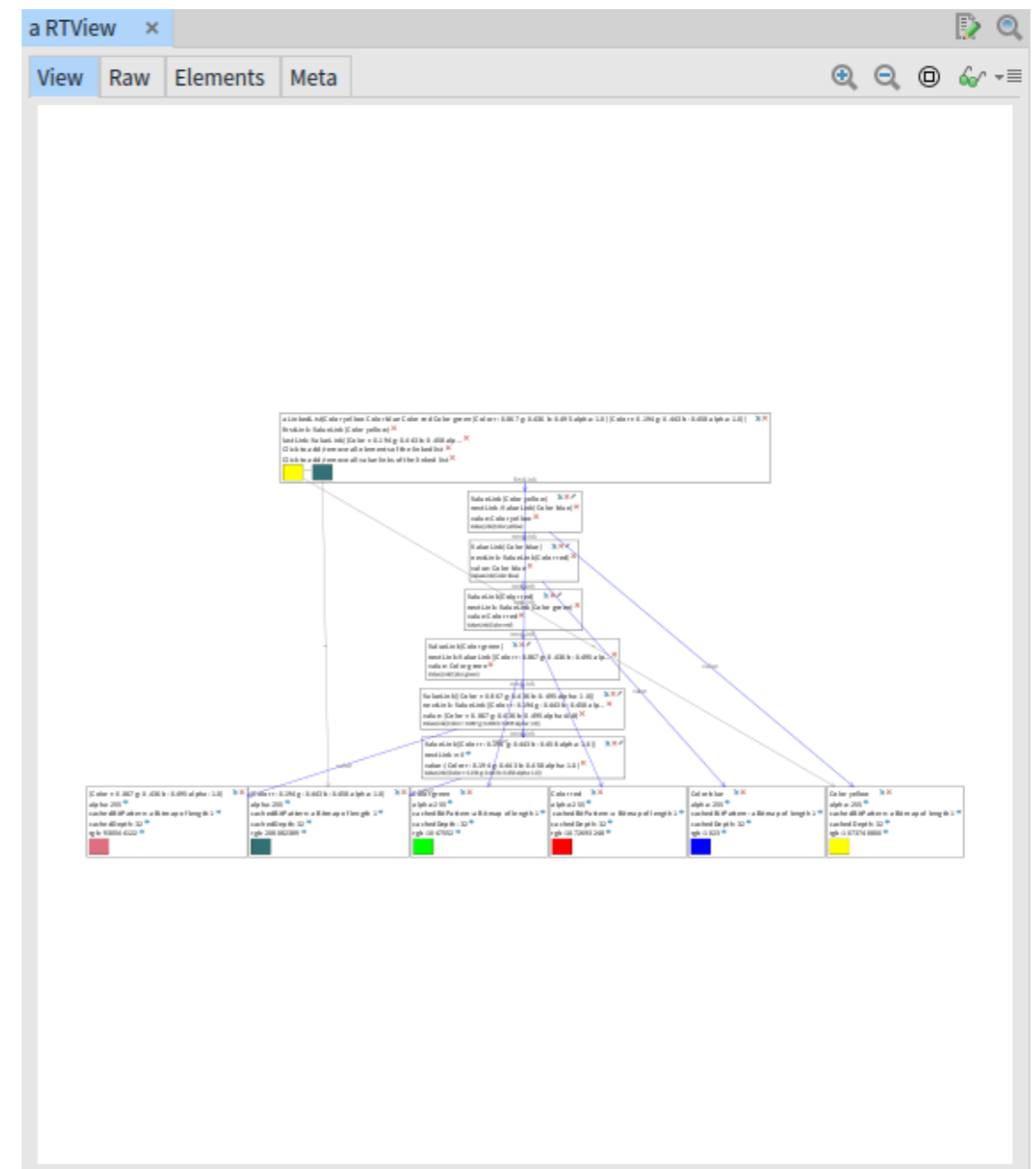


Our approach

Convey overview of multiple objects concurrently , while also conveying detailed information about individual objects

The screenshot shows a debugger window titled "a LinkedList [6 items] (Color yellow Color blue Color red Color green (Color r: 0.867 g: 0.436 b: 0.495 alpha: 1.0))". It has tabs for "Items", "Raw", and "Meta". The main area displays a tree view of the object's structure:

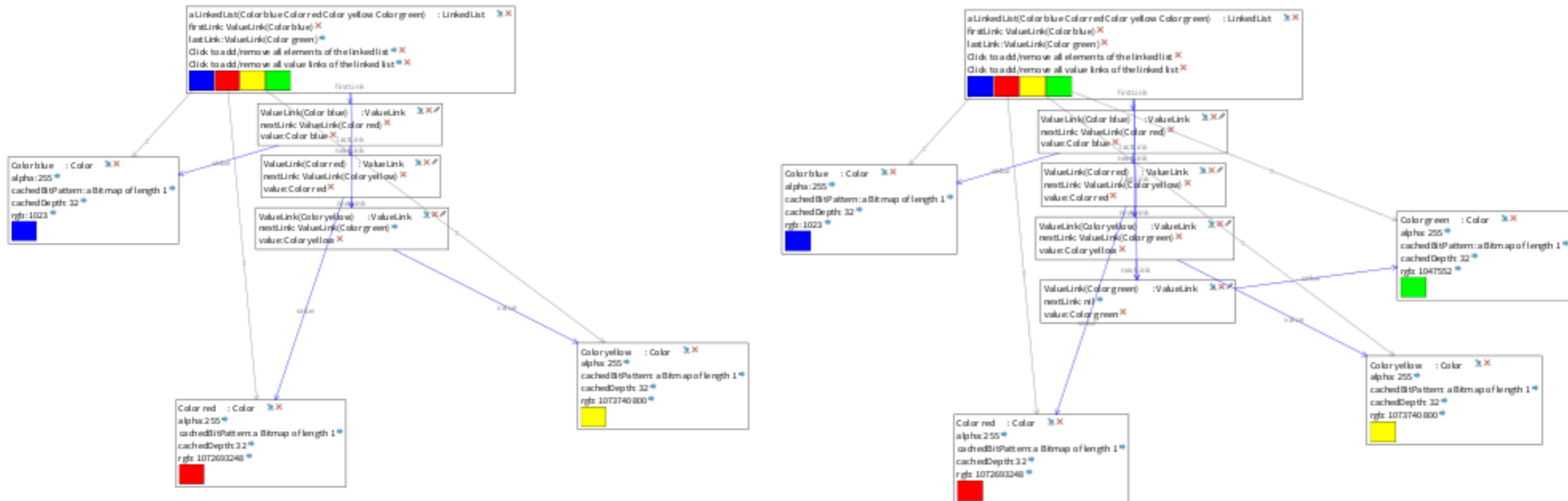
Variable	Value
{ } self	a LinkedList [6 items] (Color yellow Color blue Color red Color green (Color r: 0.867 g: 0.436 b: 0.495 alpha: 1.0))
▼ firstLink	ValueLink(Color yellow)
self	ValueLink(Color yellow)
▼ nextLink	ValueLink(Color blue)
self	ValueLink(Color blue)
▼ nextLink	ValueLink(Color red)
self	ValueLink(Color red)
▼ nextLink	ValueLink(Color green)
self	ValueLink(Color green)
▼ nextLink	ValueLink((Color r: 0.867 g: 0.436 b: 0.495 alpha: 1.0))
self	ValueLink((Color r: 0.867 g: 0.436 b: 0.495 alpha: 1.0))
▼ nextLink	ValueLink((Color r: 0.194 g: 0.443 b: 0.458 alpha: 1.0))
self	ValueLink((Color r: 0.194 g: 0.443 b: 0.458 alpha: 1.0))
▼ nextLink	nil
value	(Color r: 0.194 g: 0.443 b: 0.458 alpha: 1.0)
value	Color green
value	Color red
value	Color blue
value	Color yellow
lastLink	ValueLink((Color r: 0.194 g: 0.443 b: 0.458 alpha: 1.0))



Our approach


Enable subgraph persistency throughout renderings


Insert Color green and ValueLink Color green to graph





Our approach


Interactive visualisation


a LinkedList(Color blue Color red Color yellow Color green) : LinkedList 

firstLink: ValueLink(Color blue) 

lastLink: ValueLink(Color green) 

Click to add/remove all elements of the linked list 





Click to add/remove all value links of the linked list 










Our approach

Easy node customization

Without node customisation

```
a LinkedList(Color blue Color red Color green Color yellow)    
firstLink: ValueLink(Color blue)   
lastLink: ValueLink(Color yellow) 
```

With node customisation

```
a LinkedList(Color blue Color red Color green Color yellow)    
firstLink: ValueLink(Color blue)   
lastLink: ValueLink(Color yellow)   
Click to add/remove all elements of the linked list   
Click to add/remove all value links of the linked list   

```


Demonstration

Future Work

- Add custom nodes for core classes
- Improved layouts, class specific custom layouts
- Debugger integration

Conclusion

- Provide a tool in Pharo to visualise object sets in a graph
- Highlight relationships between objects
- Persisting subgraphs
- Direct interaction within graph
- Provide sample node customizations
- Tool makes object set inspection more efficient

