



# The *Visula* programming language and environment

Calum Grant

[calum2@visula.org](mailto:calum2@visula.org)

[www.visula.org](http://www.visula.org)



# Outline

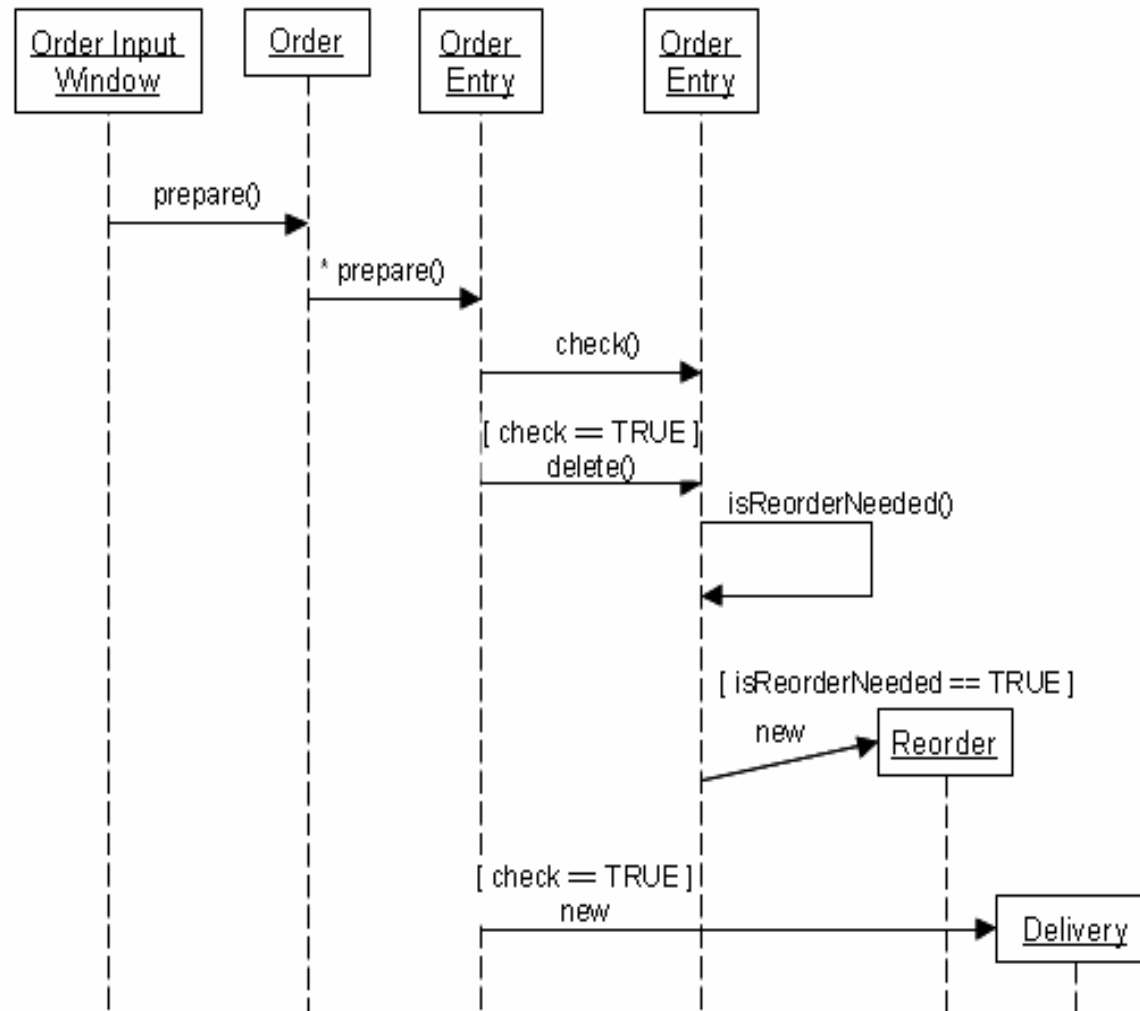
- Problem
  - Code can difficult to understand
  - Sequence diagrams help
  - General-purpose visual programming has well documented problems
  
- Proposal
  - Visula: A VPL based on sequence diagrams
  - Demo
  
- Evaluation
  - Comparison to some other notations



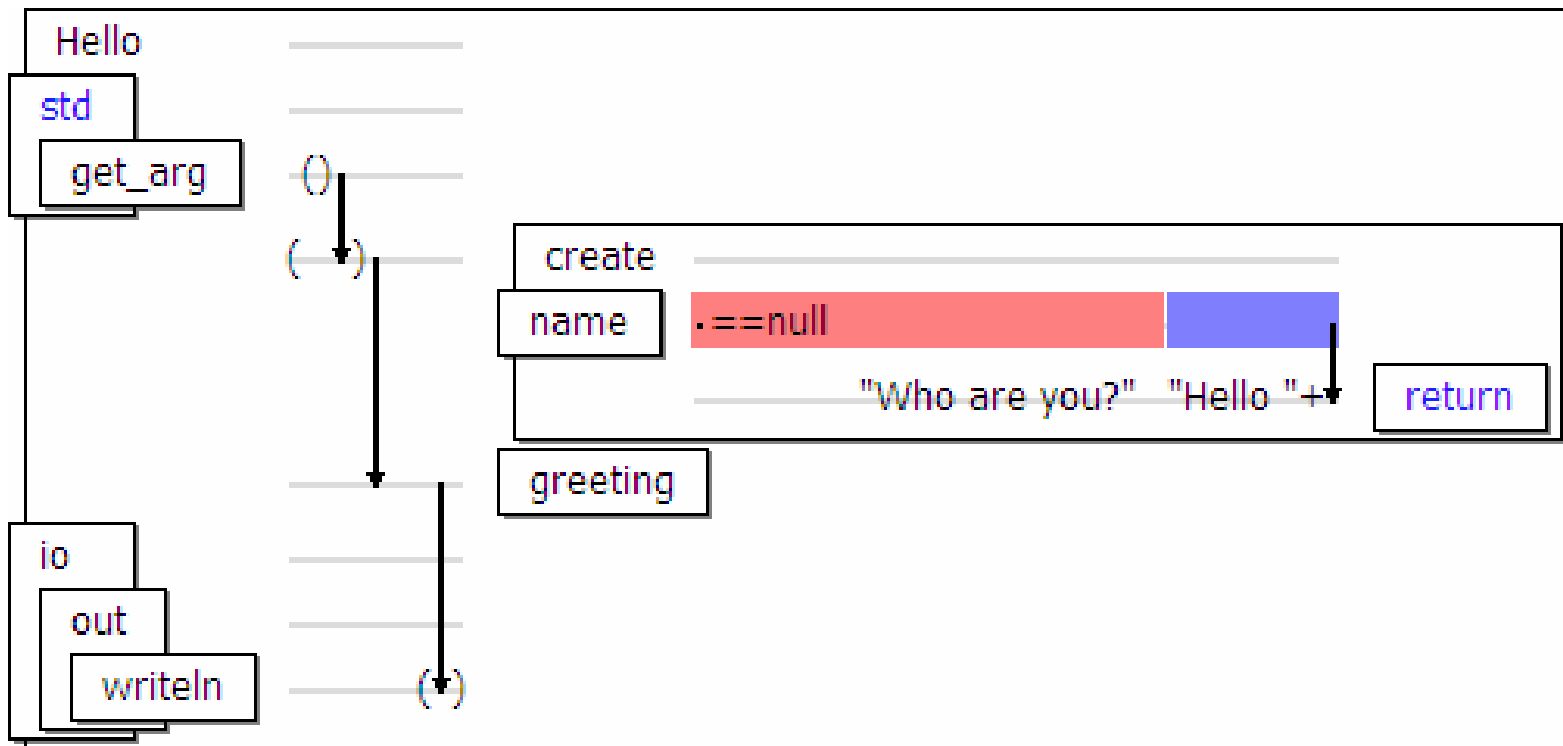
# Visula

- The first VPL based on sequence diagrams
- General purpose/scripting
- Implemented
- Can write sizable programs

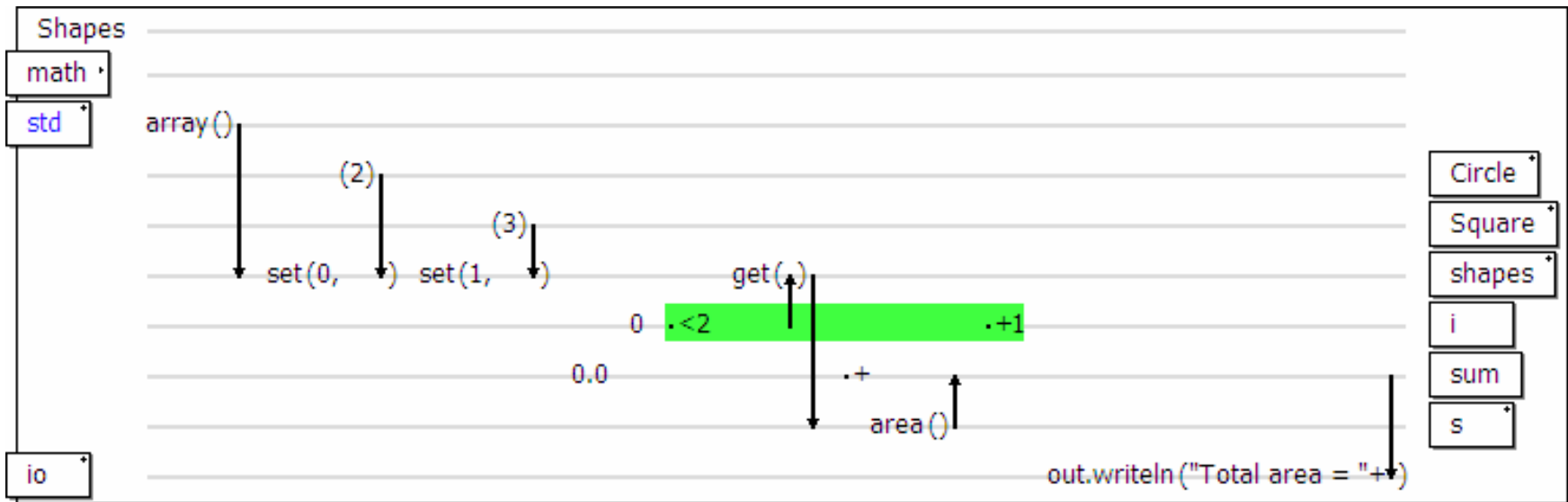
# UML sequence diagrams



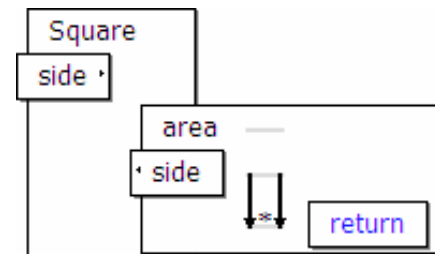
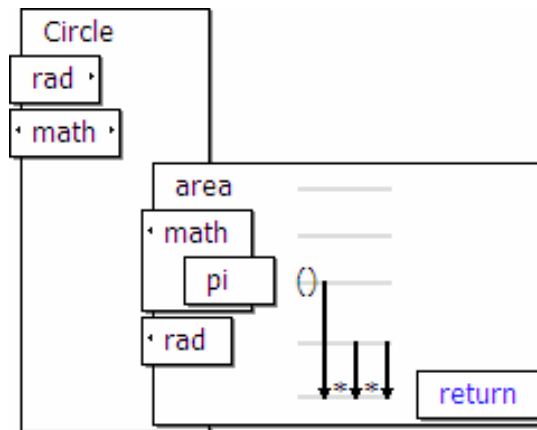
# Example 1: Hello world

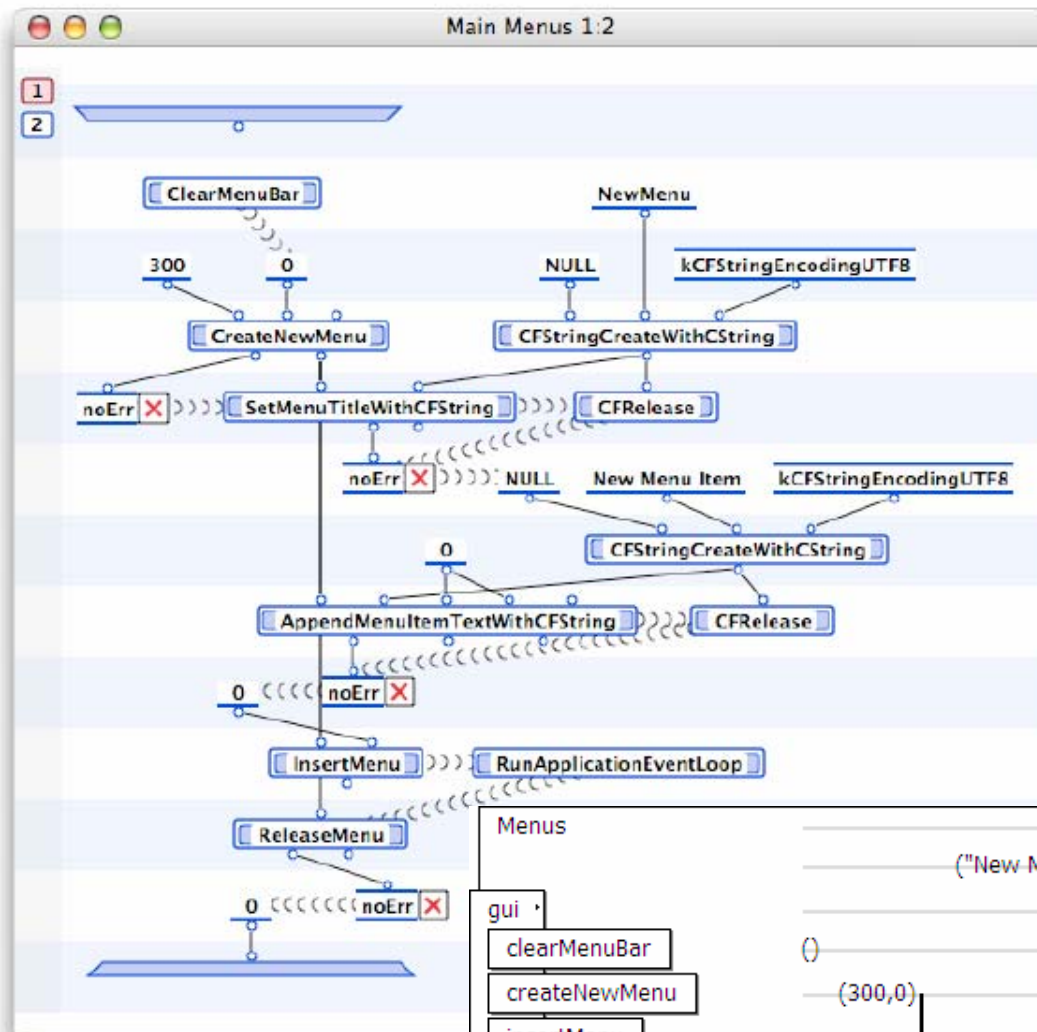


# Example 2: Shapes



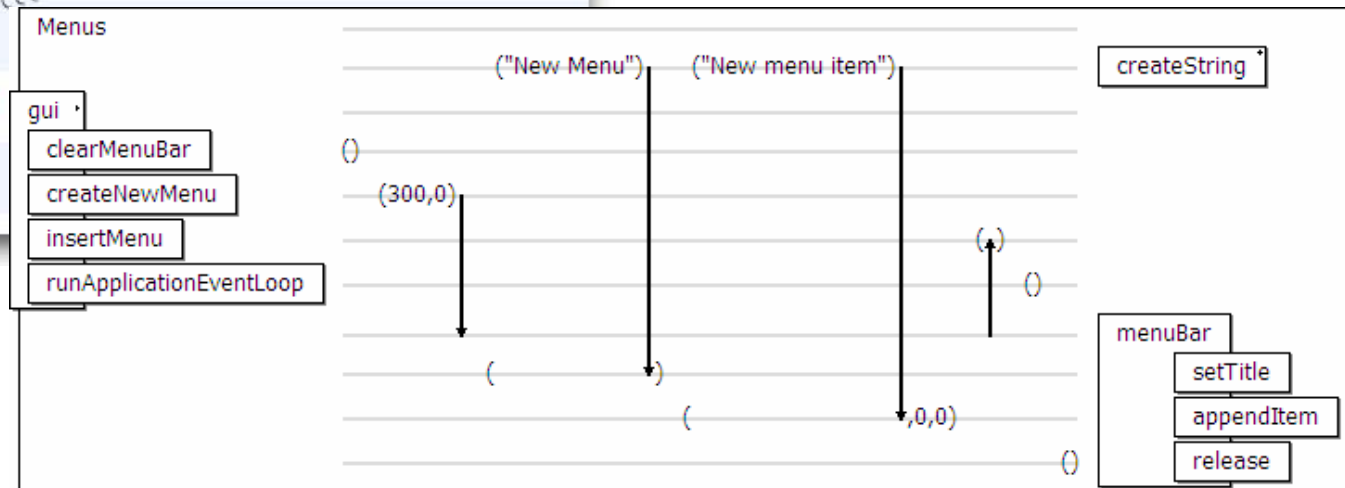
- Read from left to right



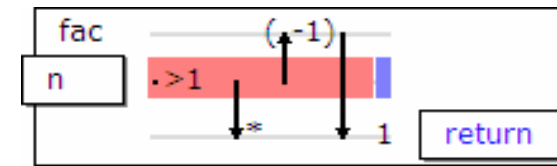
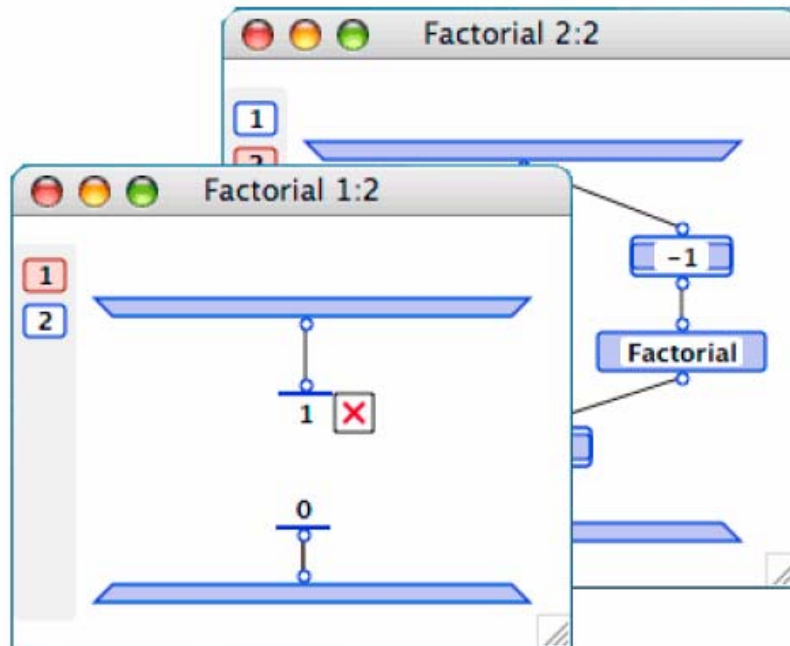


# Visula vs. Prograph

- Viscosity
- Hidden dependencies



# Visula vs. Prograph

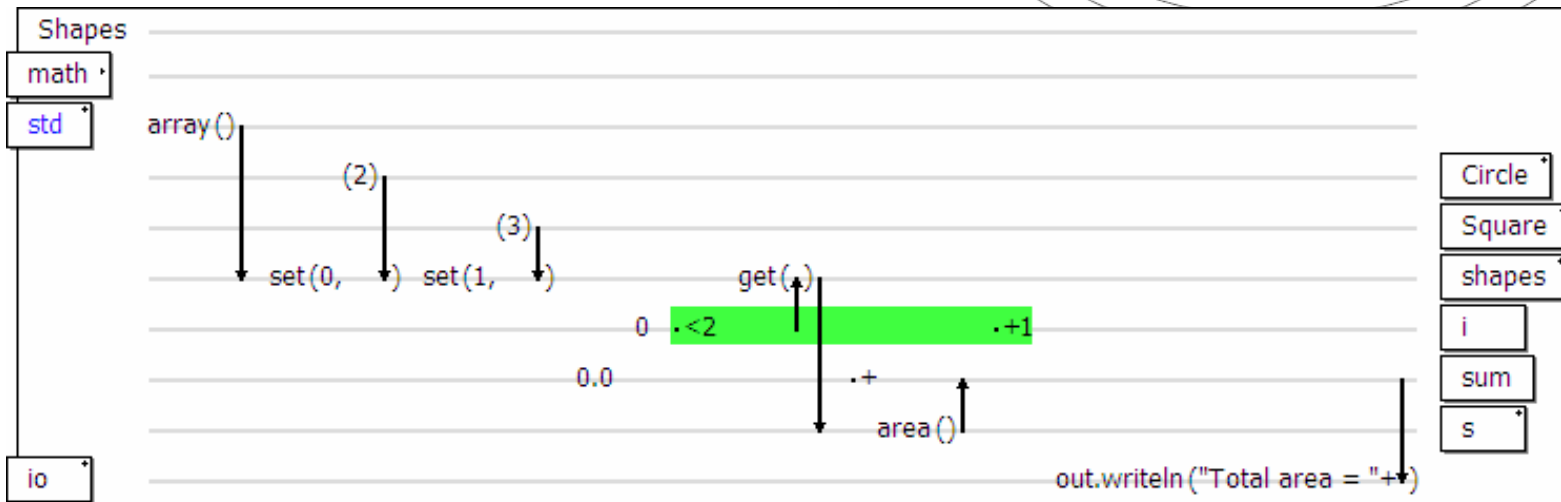
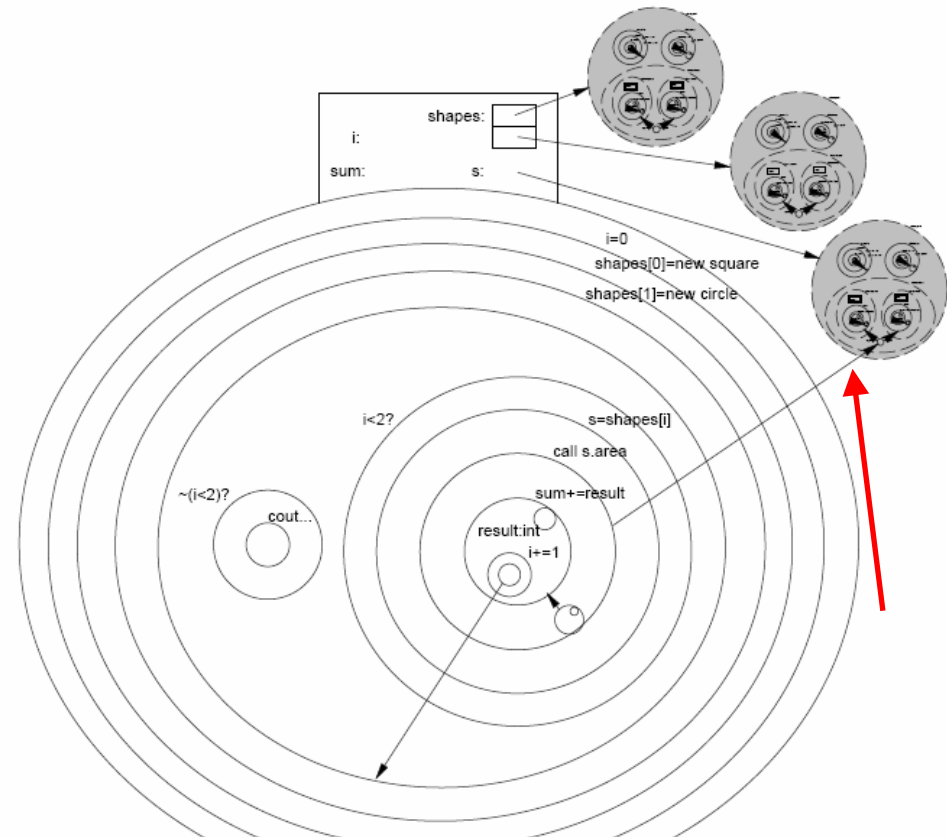


- Visibility
  - E.g. Loops?
- Diffuseness

# Visula vs. VIPR

- Diffuseness
- Dependencies
- Visibility

(what is in the grey circle?)







# Benefits over text

- Sequence diagrams show program behaviour more clearly
- Also
  - Hierarchical navigation
  - Low error-proneness
  - Low viscosity
  - High visibility
  - Low hidden dependencies



# Conclusions

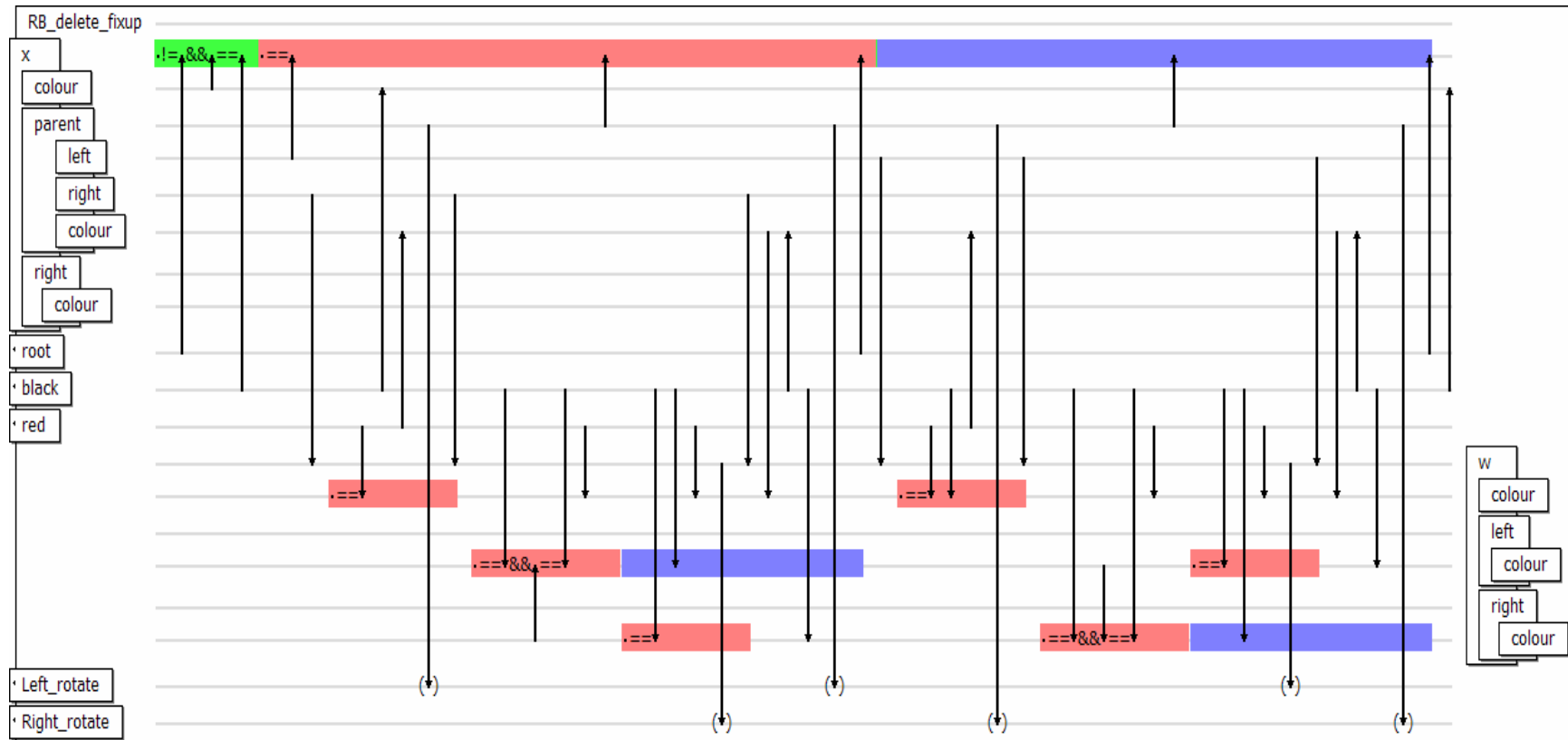
- Shown how sequence diagrams can be used as a basis for visual programming
- Informal evaluation based on cognitive dimensions
- Open source



# Future directions


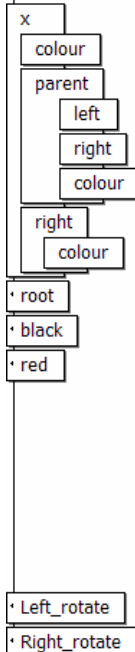
- Implementation directions
  - Cross platform, multiple selection, find/replace, performance, integration,
  
- Language directions
  - Exceptions
  - Static typing
  - Visualize concurrency
  - Wrap other languages (but how to avoid clutter??)
  
- Libraries
  - GUI, networking, database
  
- Evidence

# Why you need to break up large functions...



# Escape to text

```
RB_delete_fixup while x!=root&&x.colour==black
if x==x.parent.left
w = x.parent.right
if w.colour==red
x.colour = black
x.parent.colour = red
Left_rotate(x.parent)
w = x.parent.right
if w.left.colour==black&&w.right.colour==black
w.colour = red
x = x.parent
else
if w.right.colour==black
w.left.colour = black
w.colour = red
Right_rotate(w)
w = x.parent.right
w.colour = x.parent.colour
x.parent.colour = black
w.right.colour = black
Left_rotate(x.parent)
x = root
else
w = x.parent.left
if w.colour==red
w.colour = black
x.parent.colour = red
Right_rotate(x.parent)
w = x.parent.left
if w.right.colour==black&&w.left.colour==black
w.colour = red
x = x.parent
else
if w.left.colour==black
w.right.colour = black
w.colour = red
Left_rotate(w)
w = x.parent.left
w.colour = x.parent.colour
x.parent.colour = black
w.left.colour = black
Right_rotate(x.parent)
x = root
x.colour = black
```



# Visula vs. VIPR

