VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Mathematical and Computing Sciences

Renata Project

Interactive Visualisation as Documentation: Preliminary Report

Ewan Tempero
Robert Biddle
Stuart Marshall
Duncan Bayly

http://www.mcs.vuw.ac.nz/comp/Research/renata/

What is Software Reuse?

Using already existing artifacts from previous software development efforts in new development efforts.

- Development *with* reuse — creating new assets by using existing assets
- Development *for* reuse — how easy it is to reuse an asset

Reuse Process

- formulate query
- perform search
- retrieve component
- validate component
- adapt component
- use component

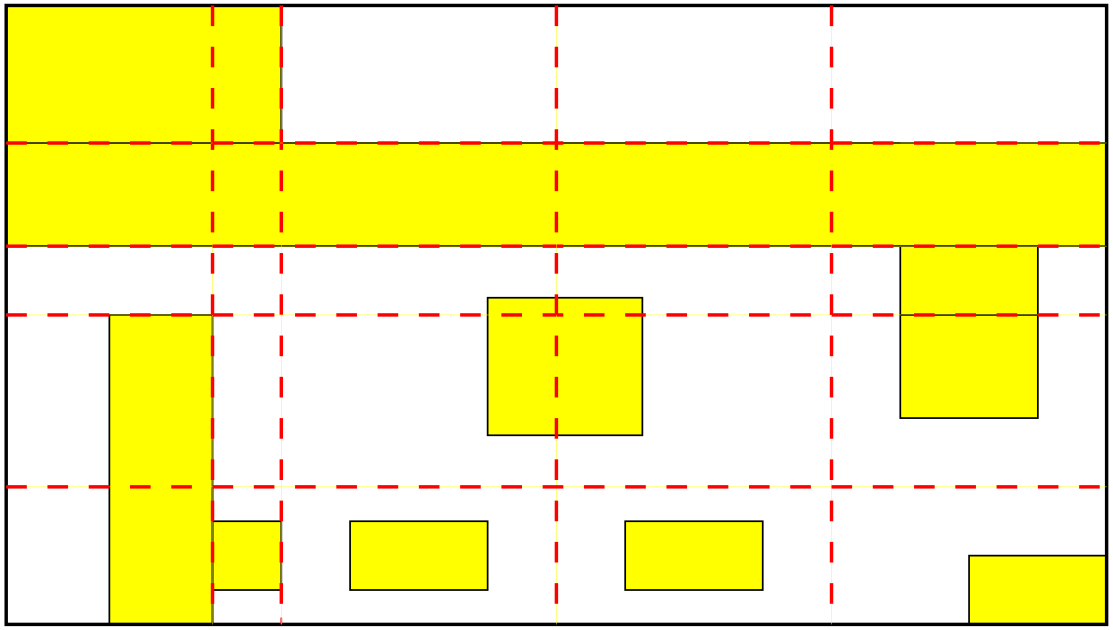
Understanding Components

- What does the component do?
- How does the component do the thing it does?
  - what methods need to be called to get the desired effect?
  - what order do they need to be called in?
  - what parameters need to be supplied?
  - what values should the parameters take?
  - why does nothing change when I change the value of that parameter?
  - why did it do *that*??!

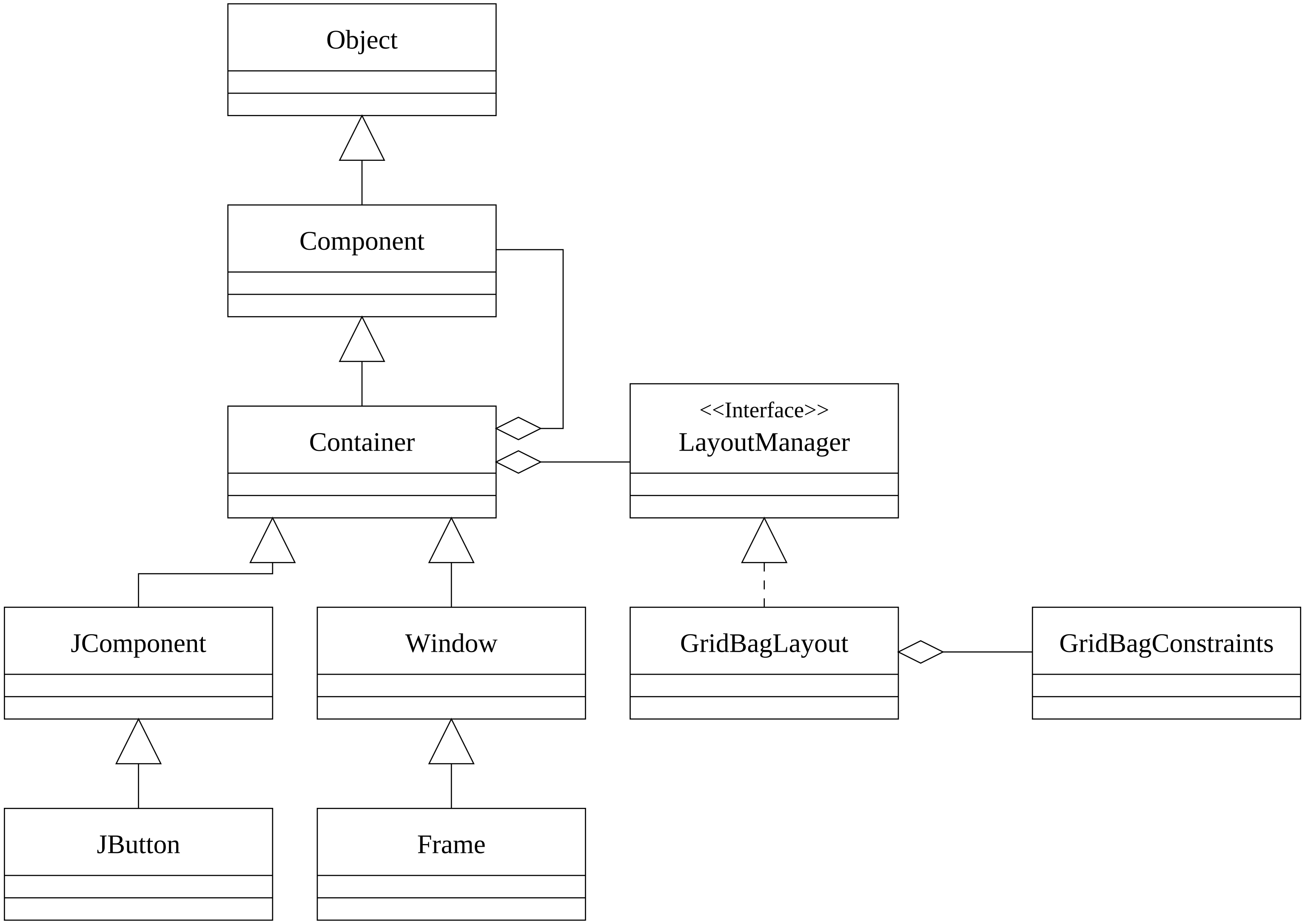
Interactive Visualisation as Documentation

- experimental programming
- interaction is more powerful than batch
- test driving software
- visualisation to help understanding

GridBagLayout: A Reusable Component



User Interface Construction in Java



GridBagConstraints

gridx, gridy: cell positions, RELATIVE

gridwidth, gridheight: specified in numbers of cells, RELATIVE, REMAINDER

anchor: CENTER, NORTH, NORTHEAST, EAST, SOUTHEAST, SOUTH, SOUTHWEST, WEST, NORTHWEST

fill: NONE, BOTH, HORIZONTAL, VERTICAL

weightx, weighty: $[0.0, 1.0]$

ipadx, ipady: in “pixels”

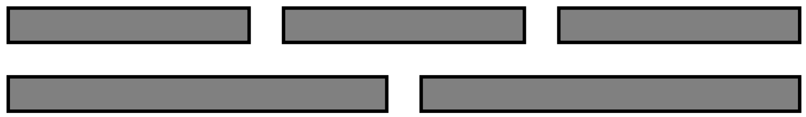
insets: top, right, bottom, left in “pixels”

Typical use

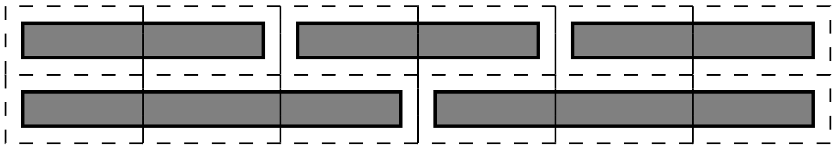
```
// In a class that extends (inherits) java.awt.Container
GridBagLayout gbl = new GridBagLayout();

// Set the layout manager for the container
setLayout(gbl);

GridBagConstraints gbc = new GridBagConstraints();
gbc.fill = GridBagConstraints.HORIZONTAL;
gbc.anchor = GridBagConstraints.NORTH;
gbc.gridx = 0;
gbc.gridy = 2;
gbc.weighty = 1.0;

// Set the constraints for the component being added to the
// container
gbl.setConstraints(component, gbc);

// Add the component to the container
add(component);
```

Typical behaviour

Goal

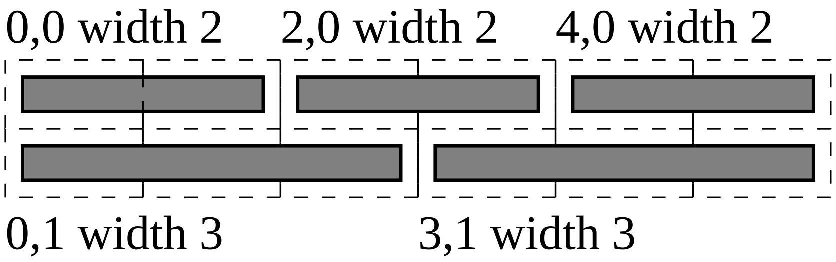


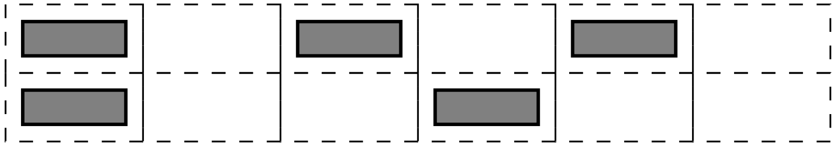
Plan



Result

Partial Explanation





GridBagDoc: Interactive Visualisation as Documentation for GridBagLayout

- tool that allows interaction with GridBagLayout
- provides visual feedback of results of actions
- uses the actual components through use of “harness” — no simulation, no instrumentation

GridBagDoc Issues

- tool encourages "improper" usage of GridBagLayout
- access to layout information
  $\Rightarrow$ observability and controllability
- what is the right interface
  $\Rightarrow$ usability
- what is the right feedback
  $\Rightarrow$ documentation

JTable

- another complicated Java widget
- lots of controls to play with
- feedback must be targeted to the programmer to be useful for documentation

Dyno

- test-drive harness
- low-cost (?) interactive visualisation as documentation
- use of JVMDI, JNI, Reflection

Future Work

- “document” non-visual components
- “bigger” components — frameworks
- user studies ($\Rightarrow$ rework tools)
- video tapes as documentation — canned visualisations
- use of java — consequences of interpretation (access, speed)
- interactive documentation vs specifications
- complicated components — frameworks

Conclusions

- basic idea seems sound — need evaulation
- controllability and observability (again)
- programmer feedback log