THE UNIVERSITY OF AUCKLAND

COMPSCI 230

Software Design and Construction

User Interface Modeling

2013-04-29

The Standish "Chaos" Report

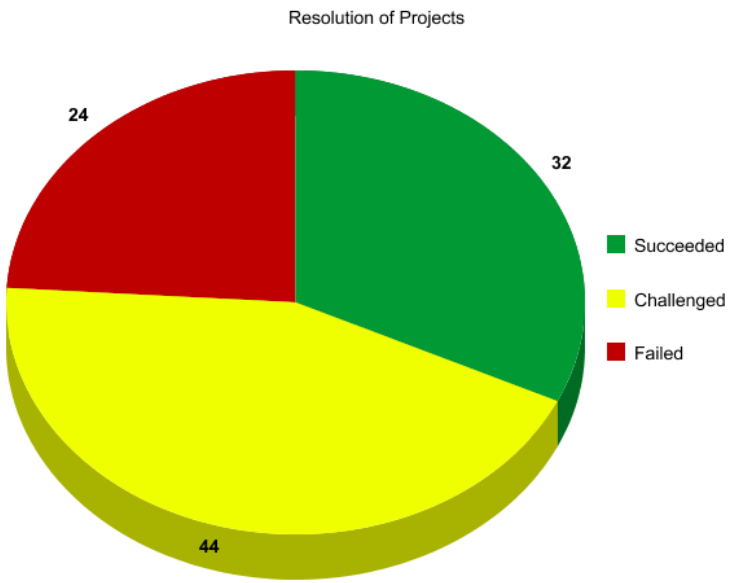
THE UNIVERSITY OF AUCKLAND

Reports on statistics about IT projects (data for 2009)

32% of all projects succeeded (delivered on time, on budget, with required features and functions)

44% are challenged (late, over budget and/or with less than the required features and functions)

24% have failed (cancelled prior to completion or delivered and never used)



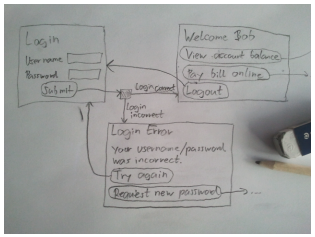
Among the suspected causes:
poor estimates
and poor planning

http://blog.standishgroup.com/

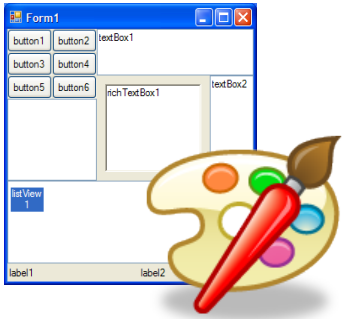
Today's Outline

1. More Swing Widgets

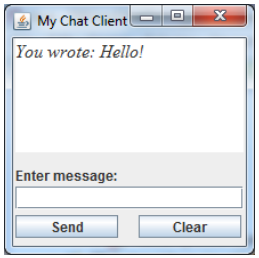
2. User Interface Modeling



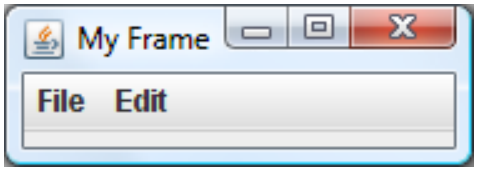
3. GUI Builders: WindowBuilder for Eclipse



4. User Interface Prototypes ("Click Dummies")



More Swing Widgets



Scrollbar Example Part 1

THE UNIVERSITY OF AUCKLAND

```
import javax.swing.*;
import java.awt.event.*;

public class ScrollBarExample extends JFrame {
  JPanel panel;
  JTextArea area;
  final JTextField field;
  JScrollPane scrollpane;

  public static void main(String[] args) {
    ScrollBarExample v = new ScrollBarExample();
    v.setVisible(true);
  }


  public ScrollBarExample() {
    // see next slide…
  }
}
```

Scrollbar Example Part 2

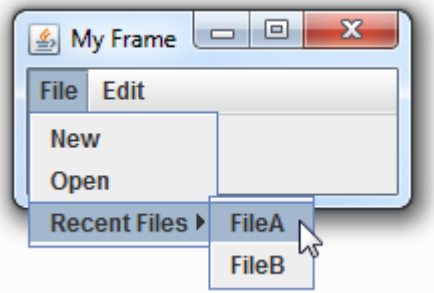
THE UNIVERSITY OF AUCKLAND

```
public ScrollBarExample() {
  super("ScrollBarExample");
  setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  setSize(100, 130);

  field = new JTextField();
  area = new JTextArea("...", 3, 7);
  scrollpane = new JScrollPane(area);
  scrollpane.getHorizontalScrollBar()
    .addAdjustmentListener(new AdjustmentListener(){
      public void adjustmentValueChanged(AdjustmentEvent e){
              field.setText("Position=" + e.getValue());
      }
  });

  panel = new JPanel();
  panel.add(field); panel.add(scrollpane);
  add(panel);
}
```

Containment Hierarchy of a Menu

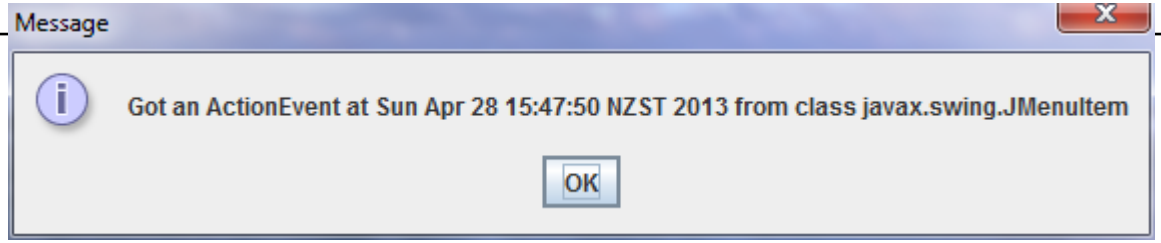
THE UNIVERSITY OF AUCKLAND

```
…
public class MenuExample extends JFrame {
    public MenuExample() {
        super("My Frame");
        setSize(200, 100);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JMenu fileMenu = new JMenu("File");
        fileMenu.add(new JMenuItem("New"));
        fileMenu.add(new JMenuItem("Open"));

        JMenu recentFilesMenu = new JMenu("Recent Files");
        recentFilesMenu.add(new JMenuItem("FileA"));
        recentFilesMenu.add(new JMenuItem("FileB"));
        fileMenu.add(recentFilesMenu);

        JMenu editMenu = new JMenu("Edit");
        editMenu.add(new JMenuItem("Undo"));
        editMenu.add(new JMenuItem("Redo"));

        JMenuBar menubar = new JMenuBar();
        menubar.add(fileMenu);
        menubar.add(editMenu);

        setJMenuBar(menubar);
}   }
```

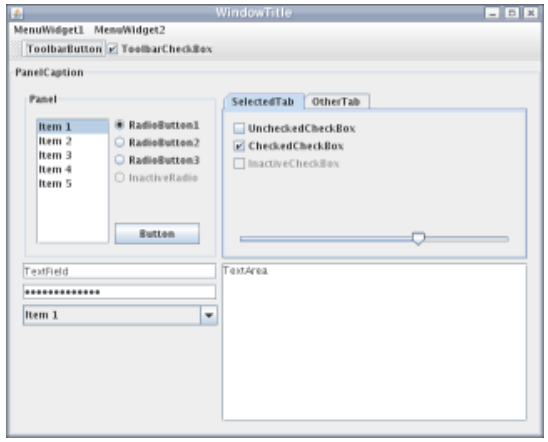


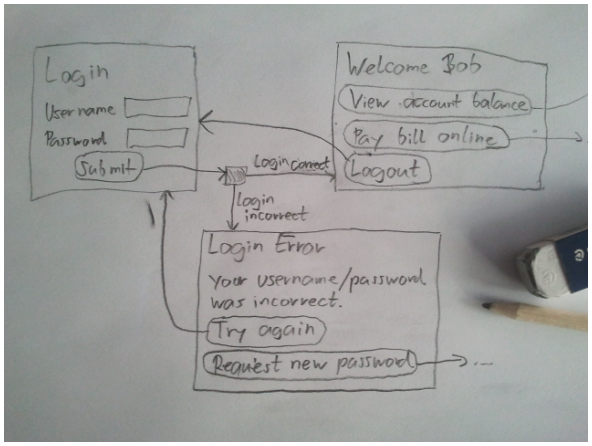
Handling Menu Events


```
...
final JFrame frame = this;
JMenuItem openMenuItem = new JMenuItem("Open");
fileMenu.add(openMenuItem);
openMenuItem.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(frame,
            "Got an ActionEvent at "
            + new Date(e.getWhen()) + " from "
            + e.getSource().getClass());
    }
});
...
```



User Interface Modeling





*

THE UNIVERSITY OF AUCKLAND

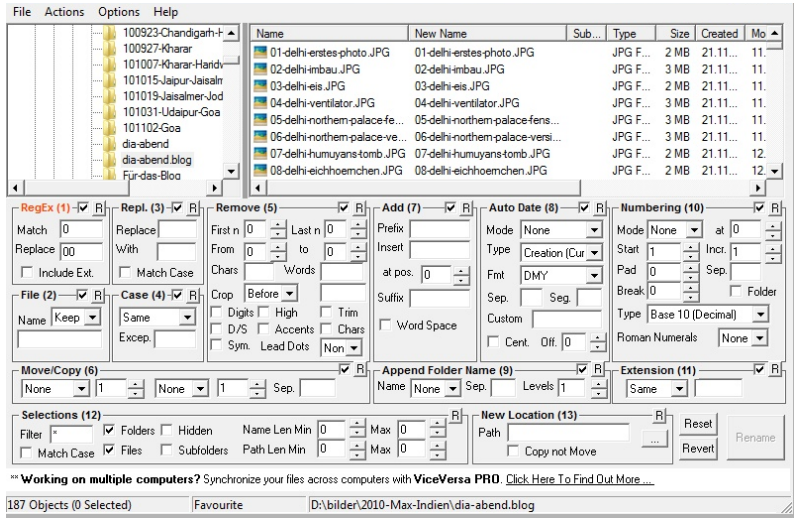
User Interfaces (UIs)

User interfaces are the interfaces between humans and computers

- **Input**: “How does the user talk to the system?“
- **Output**: “How does the system talk to the user?“
- **Interaction:** input and output between human and computer over time (HCI=Human-Computer Interaction)

The UI is a **crucial part** of a system

- Functionality is useless if users don‘t know how to use it
- Users won‘t use it if usage is cumbersome

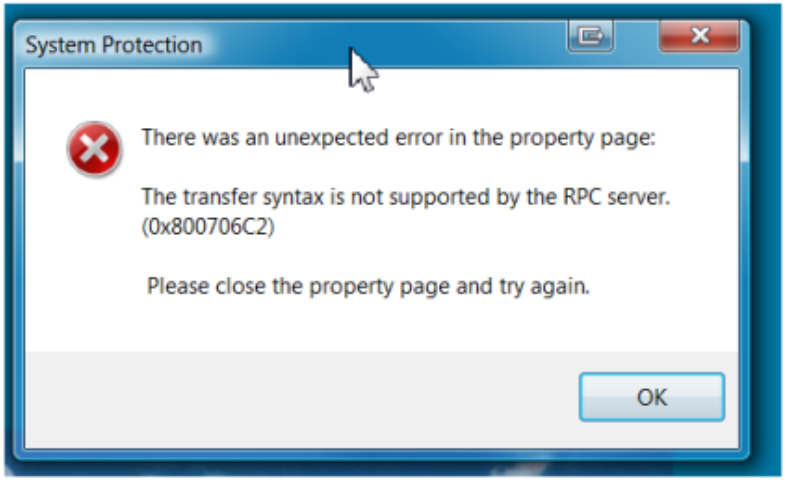


Usability

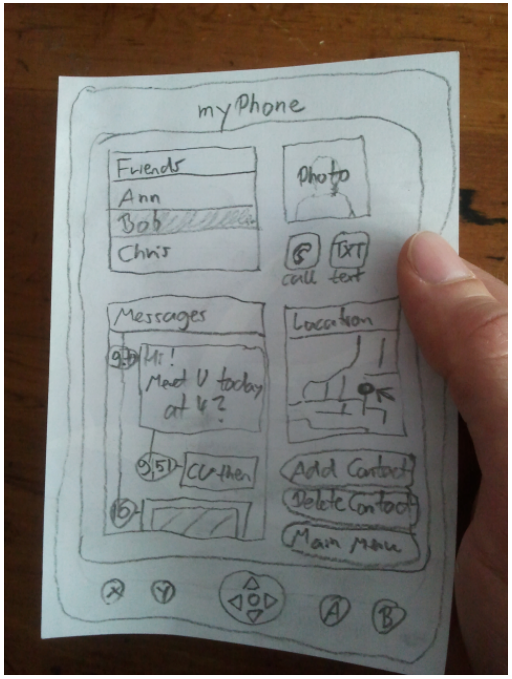
ISO 9241 definition:
"The **effectiveness**, **efficiency**, and **satisfaction** with which specified *users* achieve specified *goals* in particular *environments*."

Some **usability heuristics**
(by Jacob Nielsen http://www.useit.com/papers/heuristic)

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Help users recognize, diagnose, and recover from errors



Introduction to Modeling



Software is complex !!!
How can we deal with it?

Common solution: use a **good representation**

Model: represents certain properties
of an object in a different context

- Abstraction: reduce complexity
  by taking away unnecessary details
- Clarity: make interesting properties more visible
- Facilitate application of a methodology
- Usability (e.g. easy to create, change, understand…)

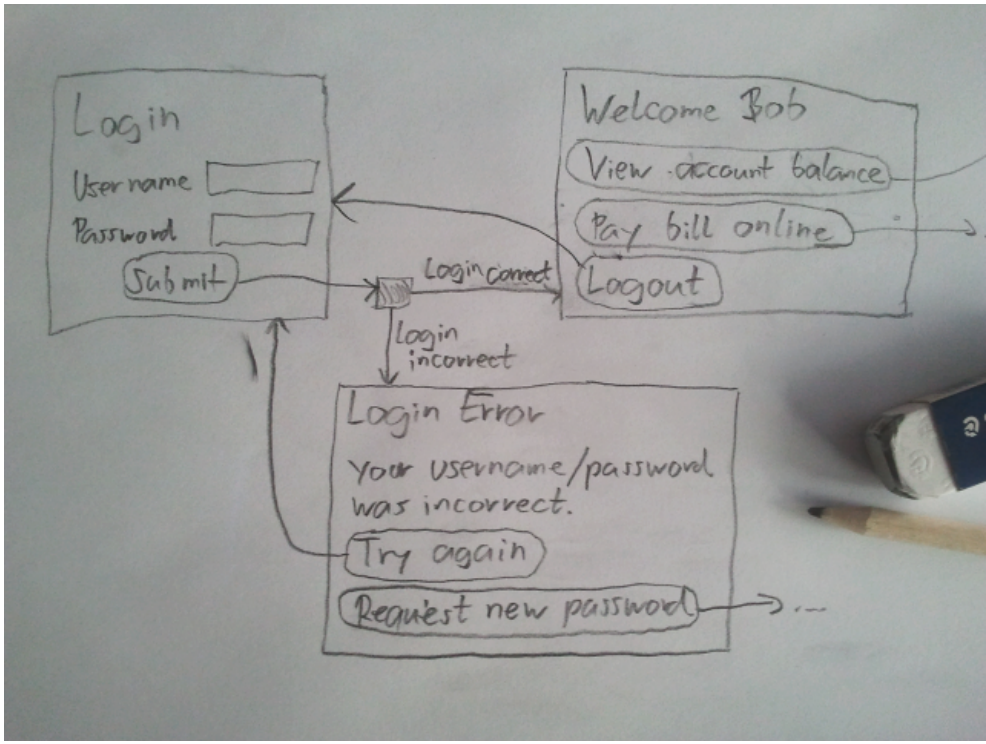
Usually many different models conceivable; different models for different purposes

Screen Diagrams

Idea: Get UI right through early user feedback
-> Use models to discuss UI with users

Screen diagrams are a simple informal model for UIs

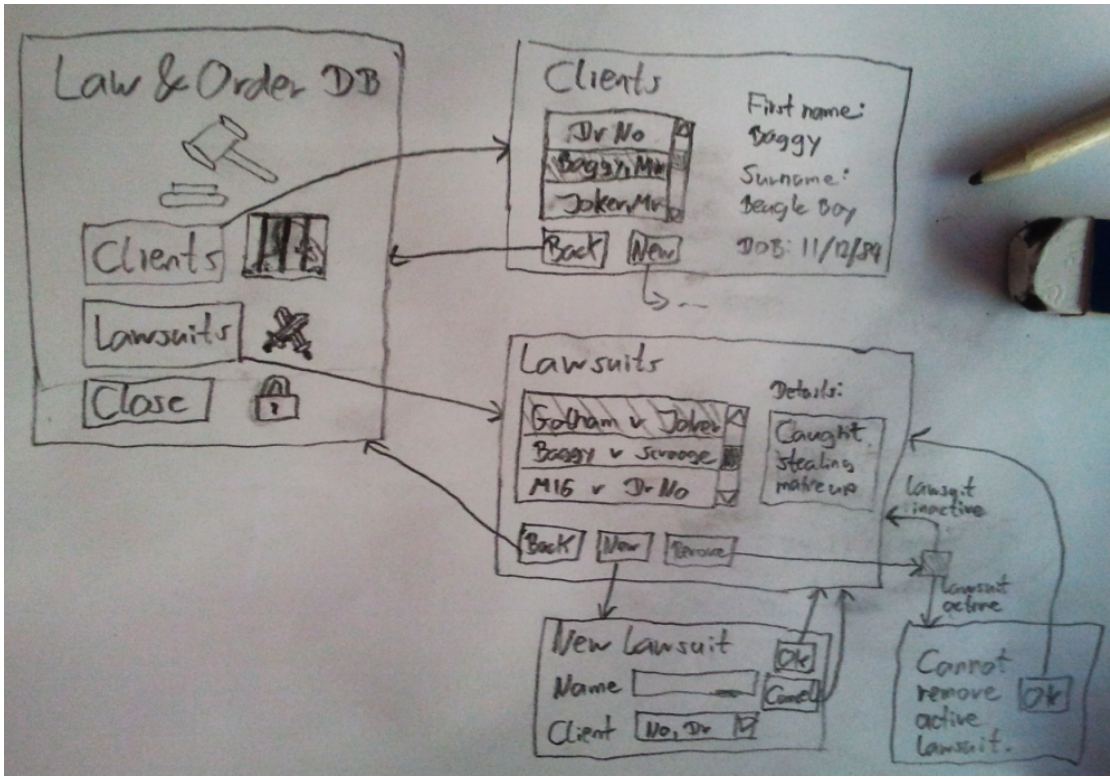
1. Draw **prototypical screens** of a system which look like real screens, with "real" data (graphical details not important)
2. Draw **arrows** from the controls of a screen to the screens that follow when the control is used (e.g. button click)
3. If multiple screens are connected to same control, insert black square signifying **conditional branch**



THE UNIVERSITY OF AUCKLAND

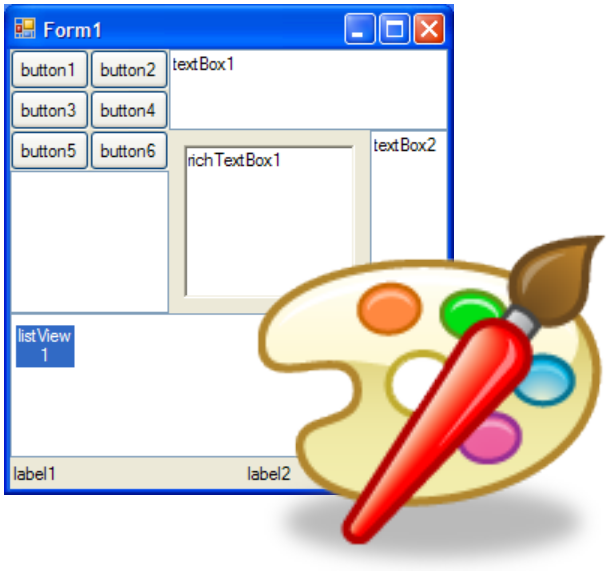
Screen Diagram Example

Create a click dummy for the following system:
A lawyer needs a program to manage clients and lawsuits. When she opens the program, she wants to see a menu with functions for listing all clients, listing all lawsuits, and closing the program. The screen that lists all the clients has a function for showing the details of a client and a function for going back to the main menu. Similarly, the screen that lists the lawsuits has a function for showing the details of a lawsuit and a function for going back.



*

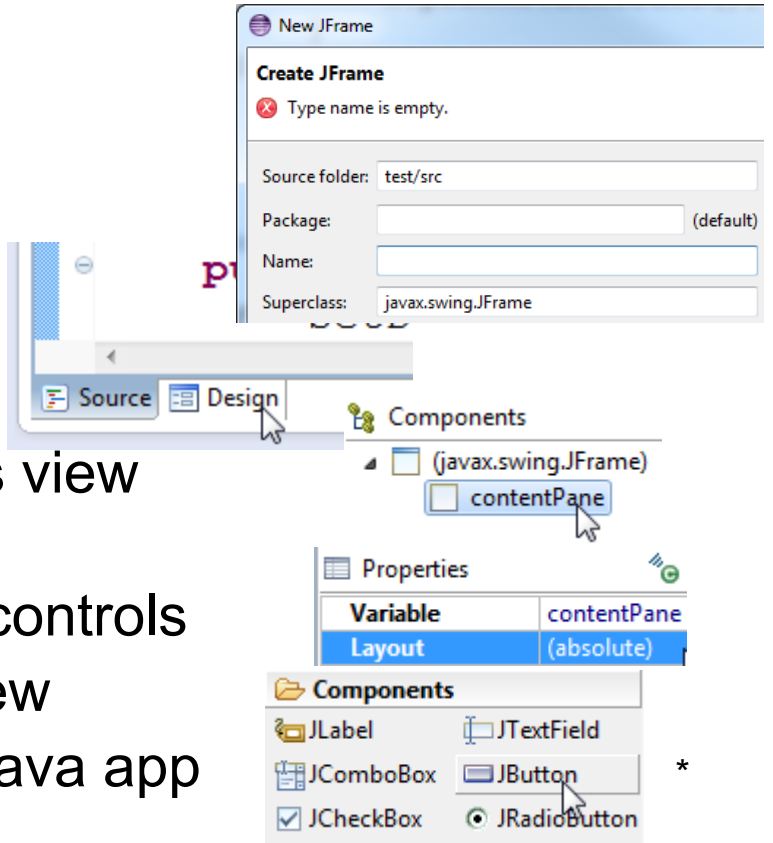
GUI Builders: WindowBuilder for Eclipse



*

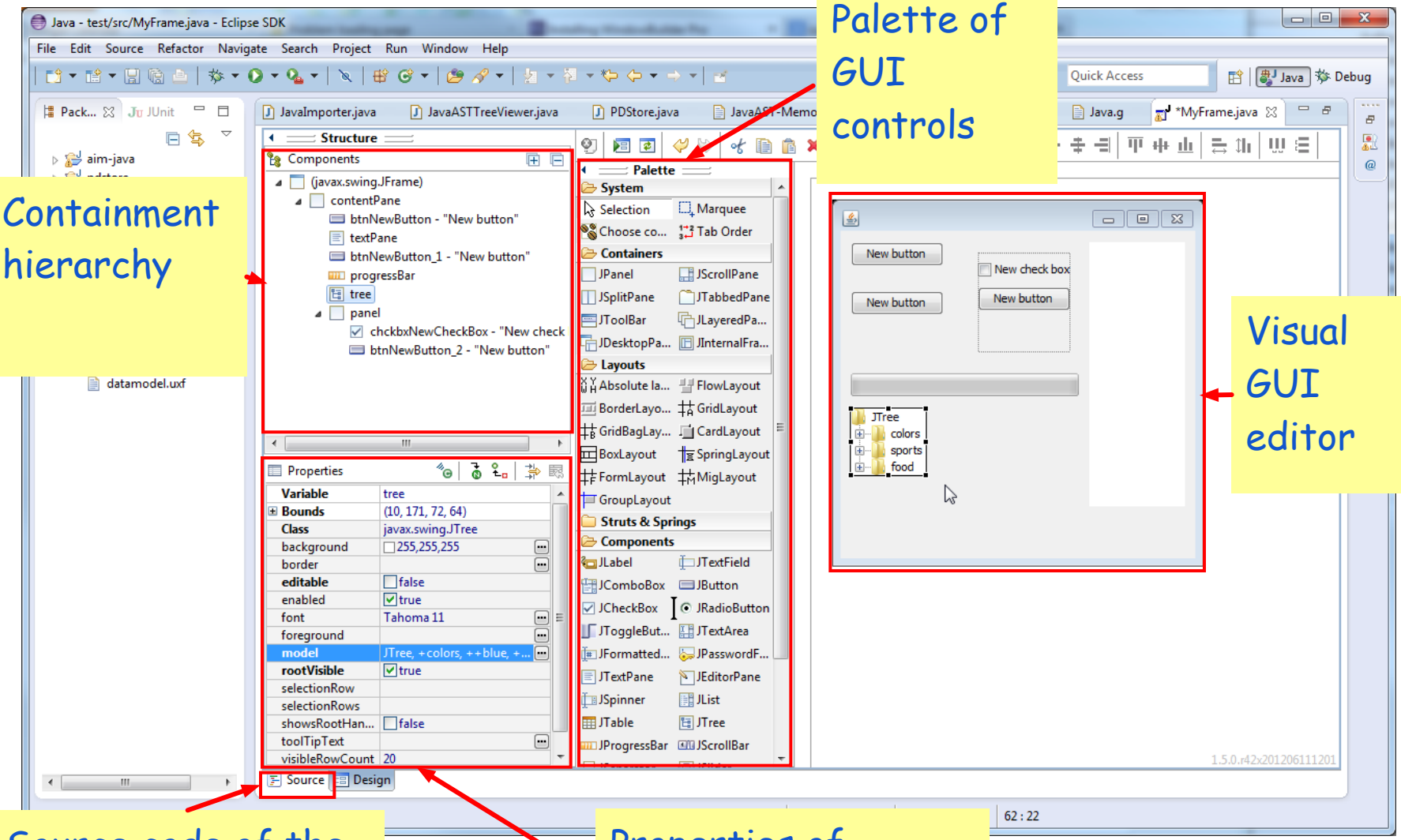
Creating a JFrame

1. Install WindowBuilder with Help -> “Install New Software” using the “update site” link for your Eclipse version from here: http://www.eclipse.org/windowbuilder/download.php (select all of “Swing Designer” and “WindowBuilder Engine” except the “WindowBuilder XML Core”)
2. Add a “JFrame” to your project using New -> Other -> WindowBuilder -> Swing Designer -> JFrame
3. Choose a package and class name
4. Switch between code and UI using the tabs at the bottom of the editor
5. Select “contentPane” in Components view
6. In Properties view: set the Layout to “(absolute)” to allow free placement of controls
7. Add components from the Palette view
8. Run the application using “Run” as Java app



WindowBuilder User Interface

THE UNIVERSITY OF AUCKLAND



Palette of GUI controls

Containment hierarchy

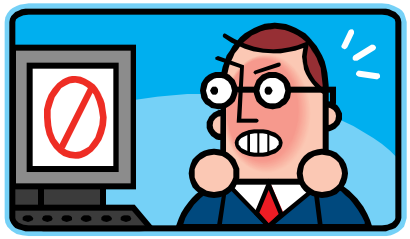
Visual GUI editor

Source code of the GUI

Properties of selected control

THE UNIVERSITY OF AUCKLAND

User Interface Prototypes ("Click Dummies")

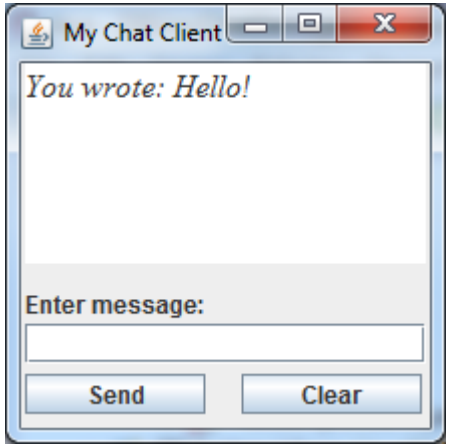


*

Click Dummies

Idea: bring screen diagrams to life with
UI mockups / UI prototypes / "click dummies"

- The user can **navigate** between the screens
- The user can see how **input & output** is done by the system
- The user can **imagine** the real system



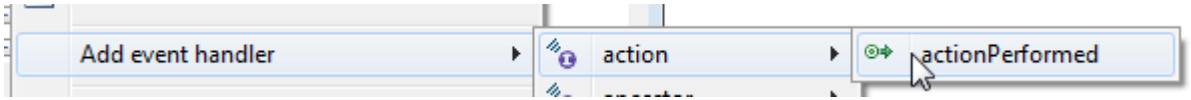
Very restricted but very easy to create

- **No functionality** implemented
- Only **hypothetical (fake) data** in the UI
- Very good for early **user testing & feedback**!!!

Opening and Closing Frames on Button Click

THE UNIVERSITY OF AUCKLAND

Add event handler by right-clicking on component and using menu



Code for opening new frame ("FrameB"):

```
final JFrame frameB = new FrameB();
btnOpenFrameB.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        frameB.setVisible(true);
}});
```

Code for closing the current frame:

```
final JFrame frameA = this;
    btnClose.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            frameA.setVisible(false);
    }});
```

Summary

- **Models** try to represent interesting aspects of a system in a clear and manageable way
- **Screen diagrams** illustrate the UI of a system
- **GUI builders** help to create UIs quickly
- **UI prototypes** (“click dummies”) can be used for early user feedback

References:

http://www.paperprototyping.com/
http://www.eclipse.org/windowbuilder/

THE UNIVERSITY OF AUCKLAND

Quiz

1. Describe a situation where using a model would be useful, and explain why (give 3 reasons).

2. Create a screen diagram for a simple media player app.

3. What is a click dummy? What is it used for?