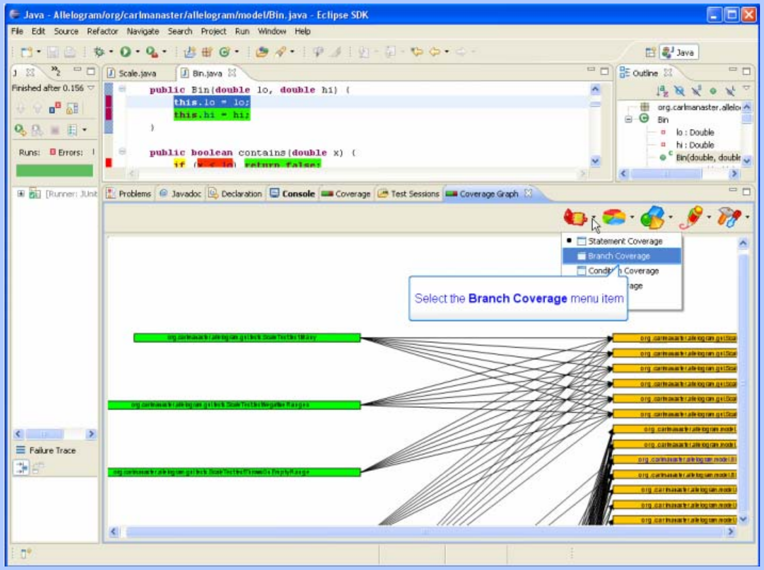
UNIVERSITY OF CALGARY

TeCReVis: A Tool for Test Coverage and Test Redundancy Visualization

Negar Koochakzadeh
Vahid Garousi

Software Quality Engineering Research Group

University of Calgary, Canada


Java - Allelogram/org/carlmanaster/allelogram/model/Bin.java - Eclipse SDK

```
public Bin(double lo, double hi) {
    this.lo = lo;
    this.hi = hi;
}

public boolean contains(double x) {
    if (x < lo) return false;
```

Finished after 0.156

Select the **Branch Coverage** menu item

SoftQual
Software Quality Engineering Research Group
Engineering Quality Software
University of Calgary, Alberta, Canada
www.softqual.ucalgary.ca

Acknowledging funding and support from:

SCHULICH School of Engineering

ALBERTA INGENUITY FUND

Alberta Innovates

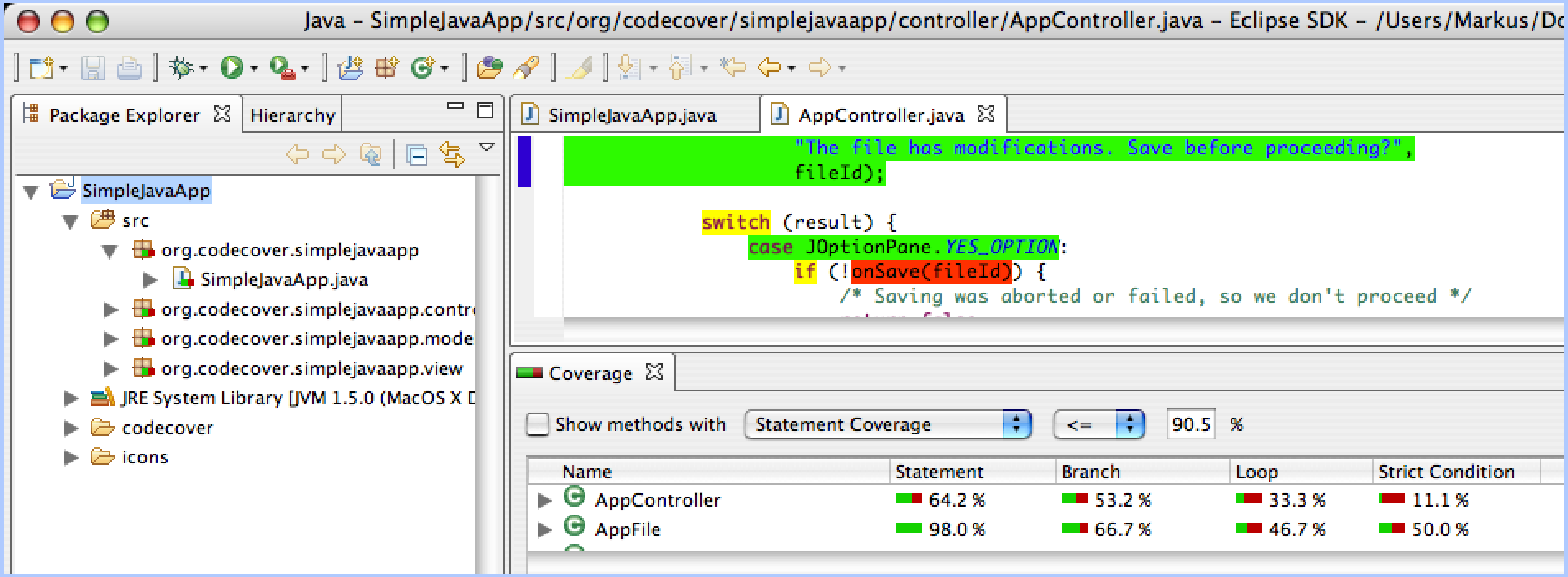
NSERC CRSNG

Talk Outline

- Existing Code Coverage Tools
- The need for Test Visualization
- 1st Feature of TeCReVis: Code Coverage Visualization
- TeCReVis - Graphical User Interface
- 2nd Feature of TeCReVis: Test Redundancy Management
- TeCReVis - Implementation Details
- Usage Scenarios of the Tool
- Availability and Demo Videos
- Q/A

Existing Code Coverage Tools

- To support automated code coverage measurement and analysis…
- test coverage values are conventionally shown in percentages and are visualized by progress-bar-like green/red boxes in the existing coverage tools
- e.g., the CodeCover plug-in for the Eclipse IDE


Java – SimpleJavaApp/src/org/codecover/simplejavaapp/controller/AppController.java – Eclipse SDK – /Users/Markus/Dc
Package Explorer
Hierarchy
SimpleJavaApp
src
org.codecover.simplejavaapp
SimpleJavaApp.java
org.codecover.simplejavaapp.contr
org.codecover.simplejavaapp.mode
org.codecover.simplejavaapp.view
JRE System Library [JVM 1.5.0 (MacOS X D
codecover
icons
SimpleJavaApp.java
AppController.java
Coverage

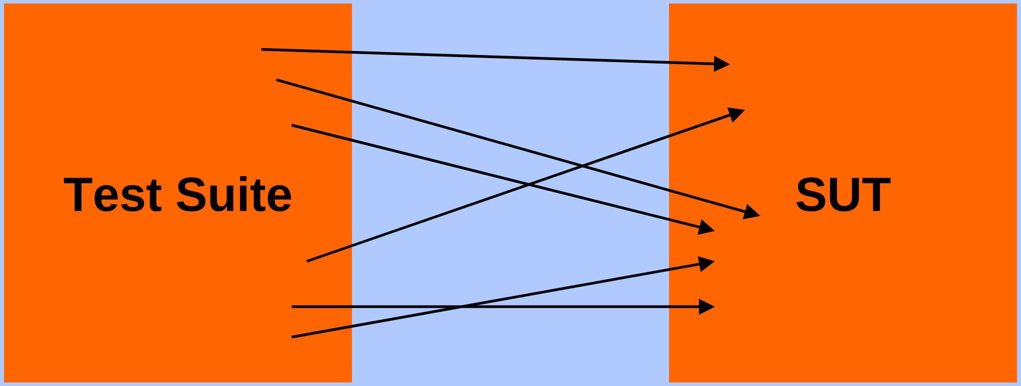
```
                    "The file has modifications. Save before proceeding?",
                    fileId);

            switch (result) {
                case JOptionPane.YES_OPTION:
                    if (!onSave(fileId)) {
                        /* Saving was aborted or failed, so we don't proceed */
```

☐ Show methods with Statement Coverage <= 90.5 %

Name	Statement	Branch	Loop	Strict Condition
AppController	64.2 %	53.2 %	33.3 %	11.1 %
AppFile	98.0 %	66.7 %	46.7 %	50.0 %

However… (The need for Test Visualization)

- However with increasing size and complexity of code bases of both systems under test and also their automated test suites (e.g., based on JUnit)
- there is a need for visualization techniques to enable testers to analyze code coverage in “higher” levels of abstraction and in holistic manners
- e.g., which packages of the SUT are covered by a specific set of test cases? Two domains…

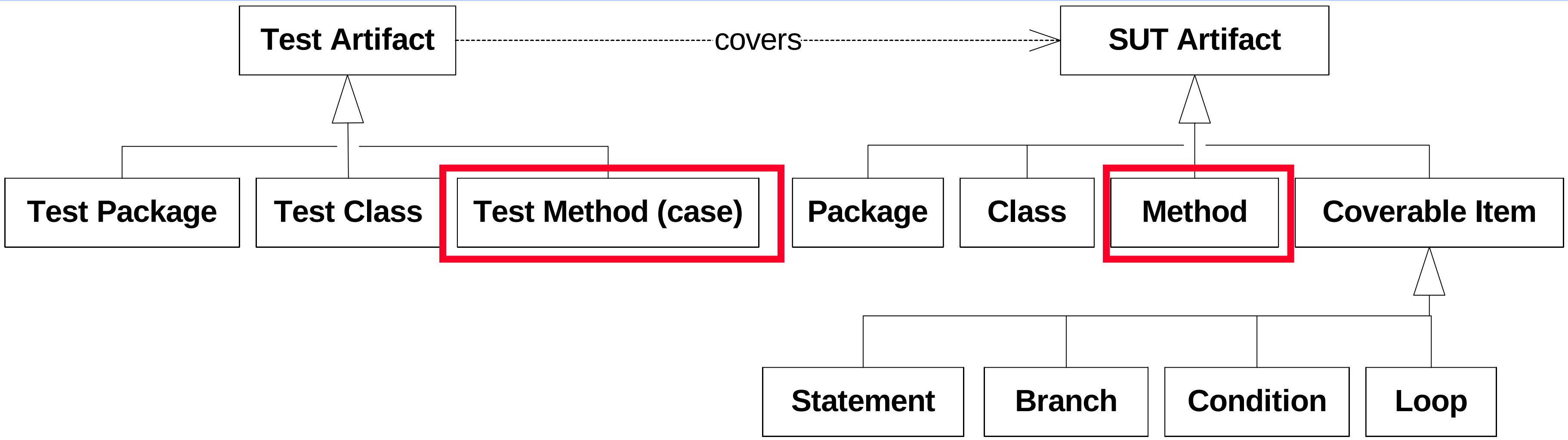


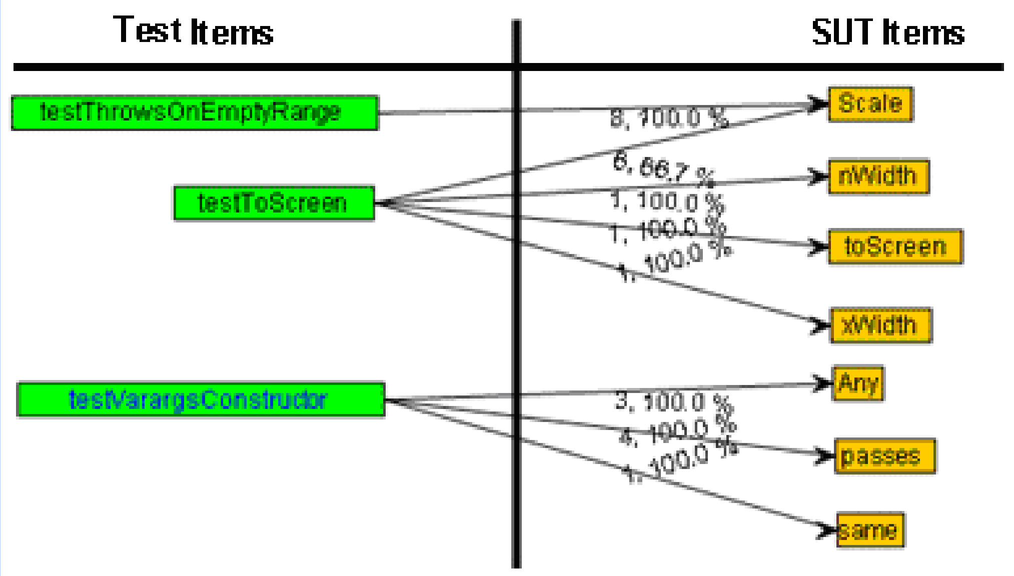
Talk Outline

- Existing Code Coverage Tools
- The need for Test Visualization
- **1st Feature of TeCReVis: Code Coverage Visualization**
- TeCReVis - Graphical User Interface
- 2nd Feature of TeCReVis: Test Redundancy Management
- TeCReVis - Implementation Details
- Usage Scenarios of the Tool
- Availability and Demo Videos
- Q/A

The Visualization Idea

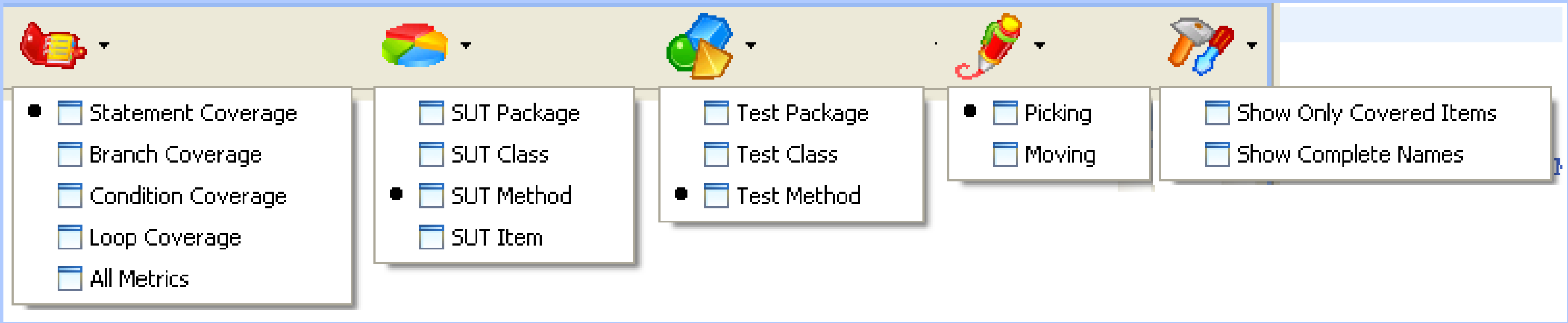
1st Feature: Test Coverage Graph (TCG)





TeCReVis - Graphical User Interface

- **Various granularities can be selected for both groups of items (e.g., method, class or package).**



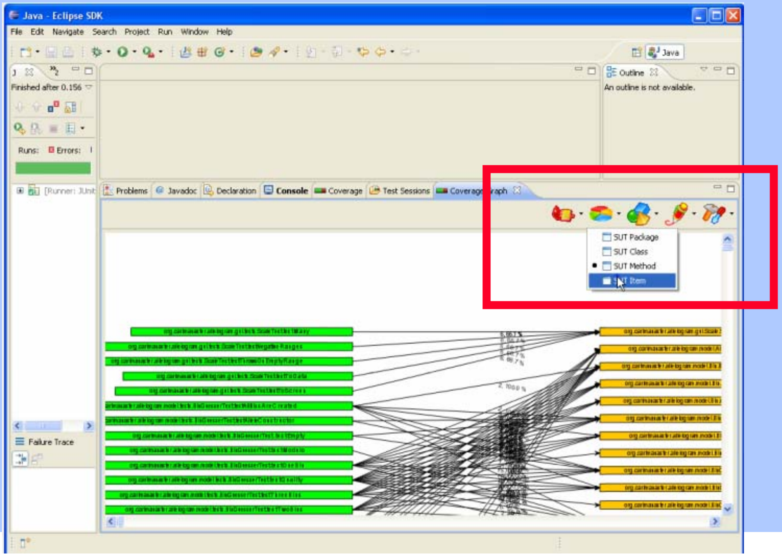
- ● □ Statement Coverage
- □ Branch Coverage
- □ Condition Coverage
- □ Loop Coverage
- □ All Metrics

- □ SUT Package
- □ SUT Class
- ● □ SUT Method
- □ SUT Item

- □ Test Package
- □ Test Class
- ● □ Test Method

- ● □ Picking
- □ Moving

- □ Show Only Covered Items
- □ Show Complete Names


Java - Eclipse SDK
File Edit Navigate Search Project Run Window Help
Java
Outline
An outline is not available.
Finished after 0.156
Runs: Errors:
Problems Javadoc Declaration Console Coverage Test Sessions Coverage Graph
SUT Package
SUT Class
SUT Method
SUT Item
Failure Trace

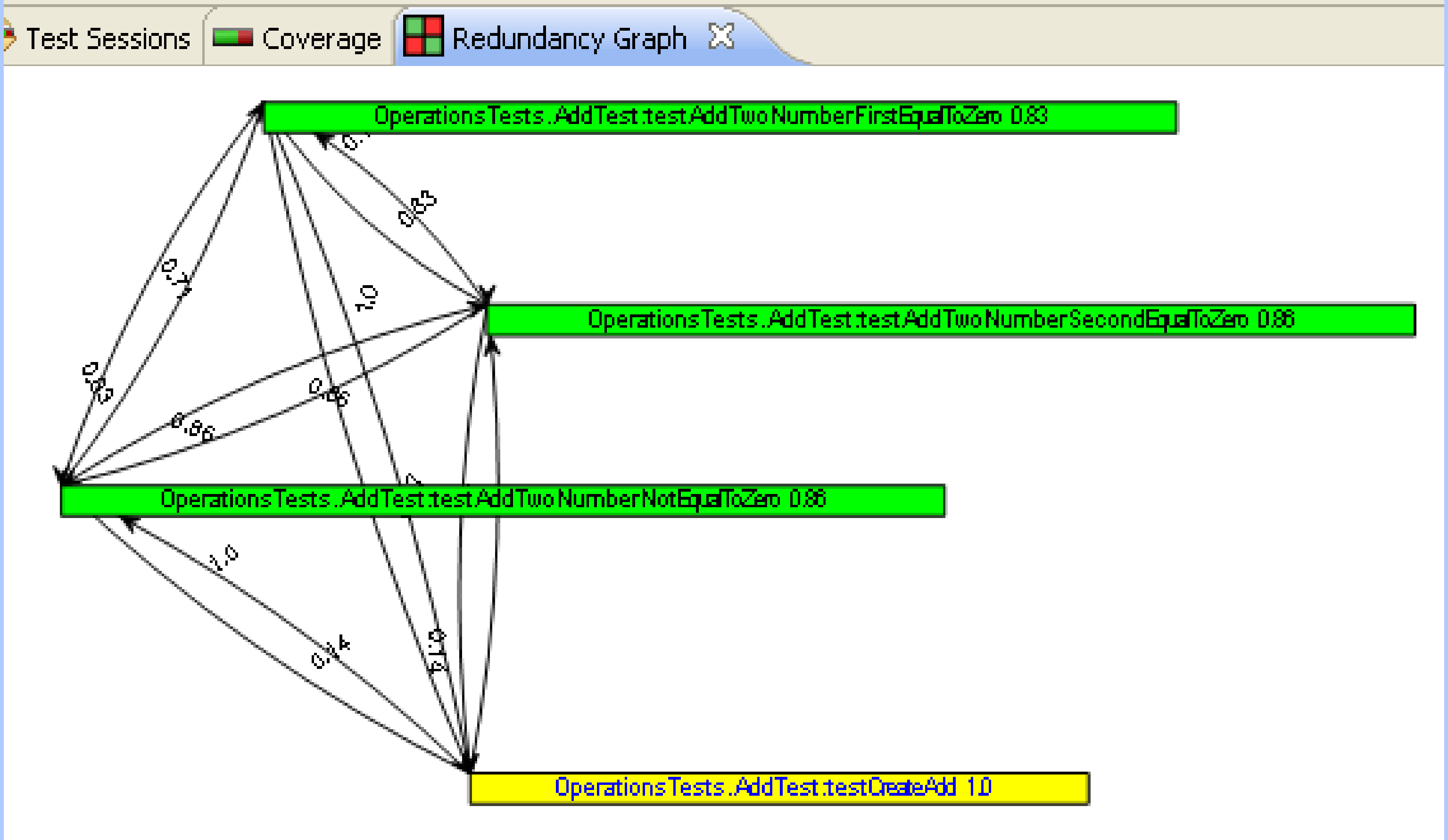
Talk Outline

- Existing Code Coverage Tools
- The need for Test Visualization
- 1st Feature of TeCReVis: Code Coverage Visualization
- TeCReVis - Graphical User Interface
- **2nd Feature of TeCReVis: Test Redundancy Management**
- TeCReVis - Implementation Details
- Usage Scenarios of the Tool
- Availability and Demo Videos
- Q/A

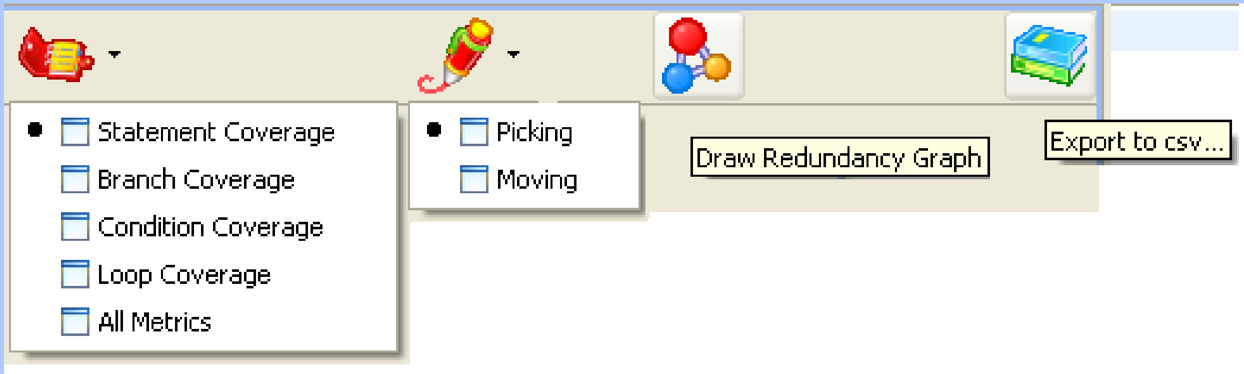
2nd Feature of TeCReVis

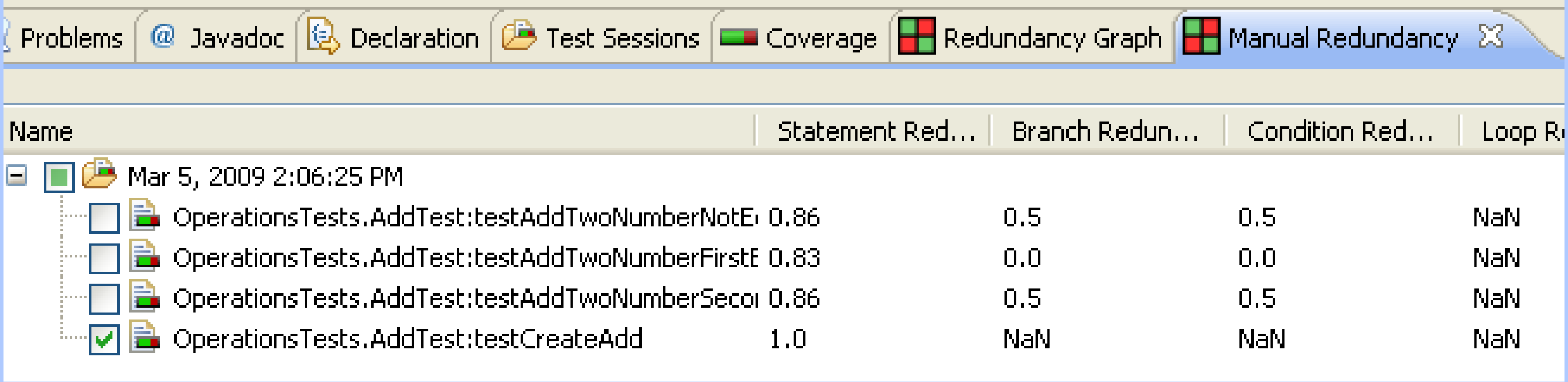
- **TeCReVis: A Tool for Test Coverage and Test Redundancy Visualization**
- **Test Redundancy Graph (TRG)**
- **The test redundancy metrics are defined in an earlier paper***

N. Koochakzadeh, V. Garousi, and F. Maurer, "Test Redundancy Measurement Based on Coverage Information: Evaluation and Lessons Learned," in *Proc. of Int. Conf. on Soft. Testing, Verification, and Validation (ICST)*, 2009.


Test Sessions
Coverage
Redundancy Graph
OperationsTests.AddTest.testAddTwoNumberFirstEqualToZero 0.83
OperationsTests.AddTest.testAddTwoNumberSecondEqualToZero 0.86
OperationsTests.AddTest.testAddTwoNumberNotEqualToZero 0.86
OperationsTests.AddTest.testCreateAdd 1.0
0.83
0.7
1.0
0.83
0.86
0.86
1.0
0.14
0.14

Test Redundancy Management using TeCReVis





Name	Statement Red...	Branch Redun...	Condition Red...	Loop R
⊟ ■ Mar 5, 2009 2:06:25 PM				
□ OperationsTests.AddTest:testAddTwoNumberNotE	0.86	0.5	0.5	NaN
□ OperationsTests.AddTest:testAddTwoNumberFirstE	0.83	0.0	0.0	NaN
□ OperationsTests.AddTest:testAddTwoNumberSeco	0.86	0.5	0.5	NaN
☑ OperationsTests.AddTest:testCreateAdd	1.0	NaN	NaN	NaN

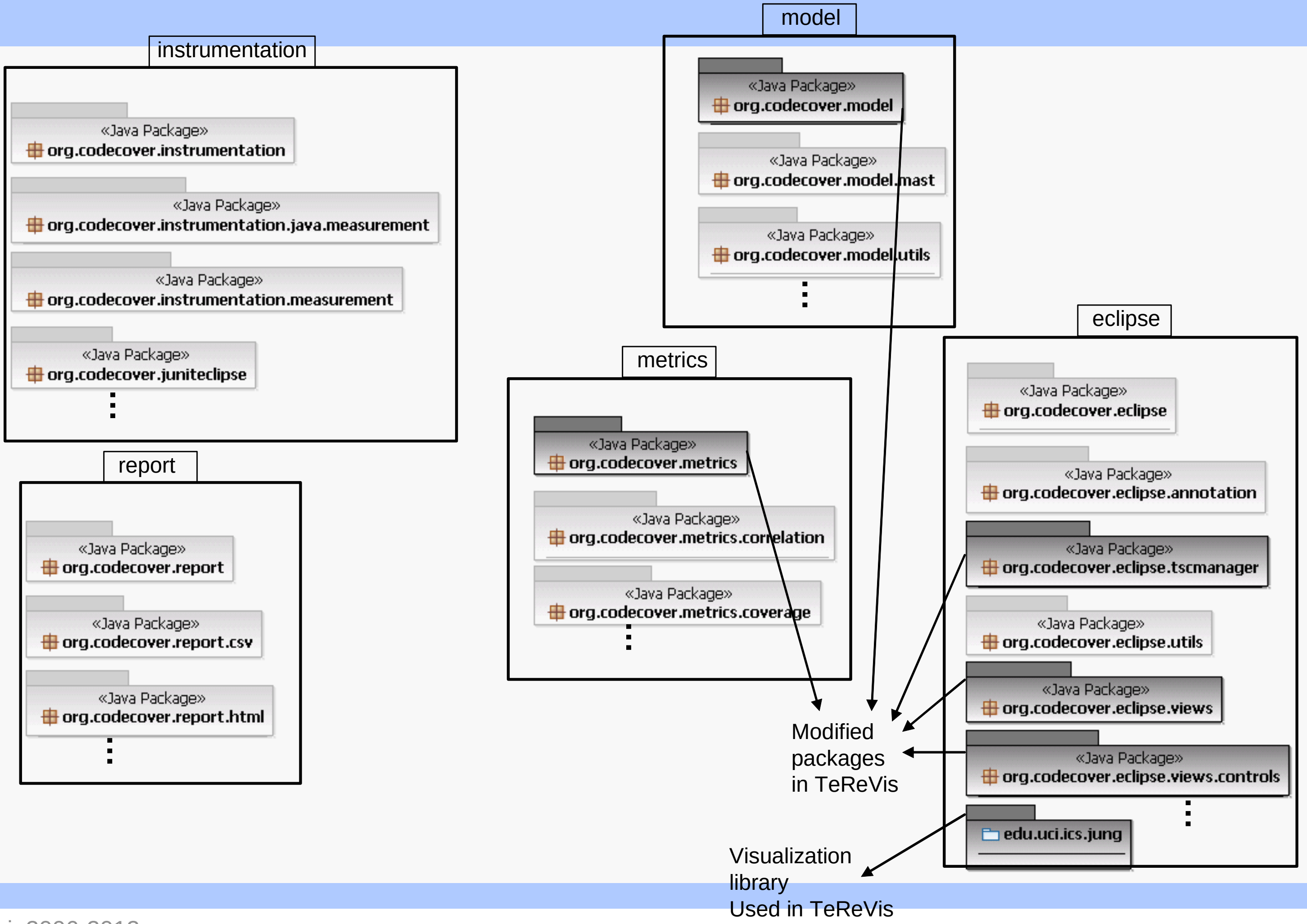
For details see…N. Koochakzadeh and V. Garousi “A Tester-Assisted Methodology for Test Redundancy Detection”, Journal on Advances in Software Engineering, Special Issue on Software Test Automation, 2010: pp. 1-13.

Talk Outline

- Existing Code Coverage Tools
- The need for Test Visualization
- 1st Feature of TeCReVis: Code Coverage Visualization
- TeCReVis - Graphical User Interface
- 2nd Feature of TeCReVis: Test Redundancy Management
- **TeCReVis - Implementation Details**
- Usage Scenarios of the Tool
- Availability and Demo Videos
- Q/A

TeCReVis - Implementation Details

- **Simplified package diagram of CodeCover and the modified parts in TeCReVis**

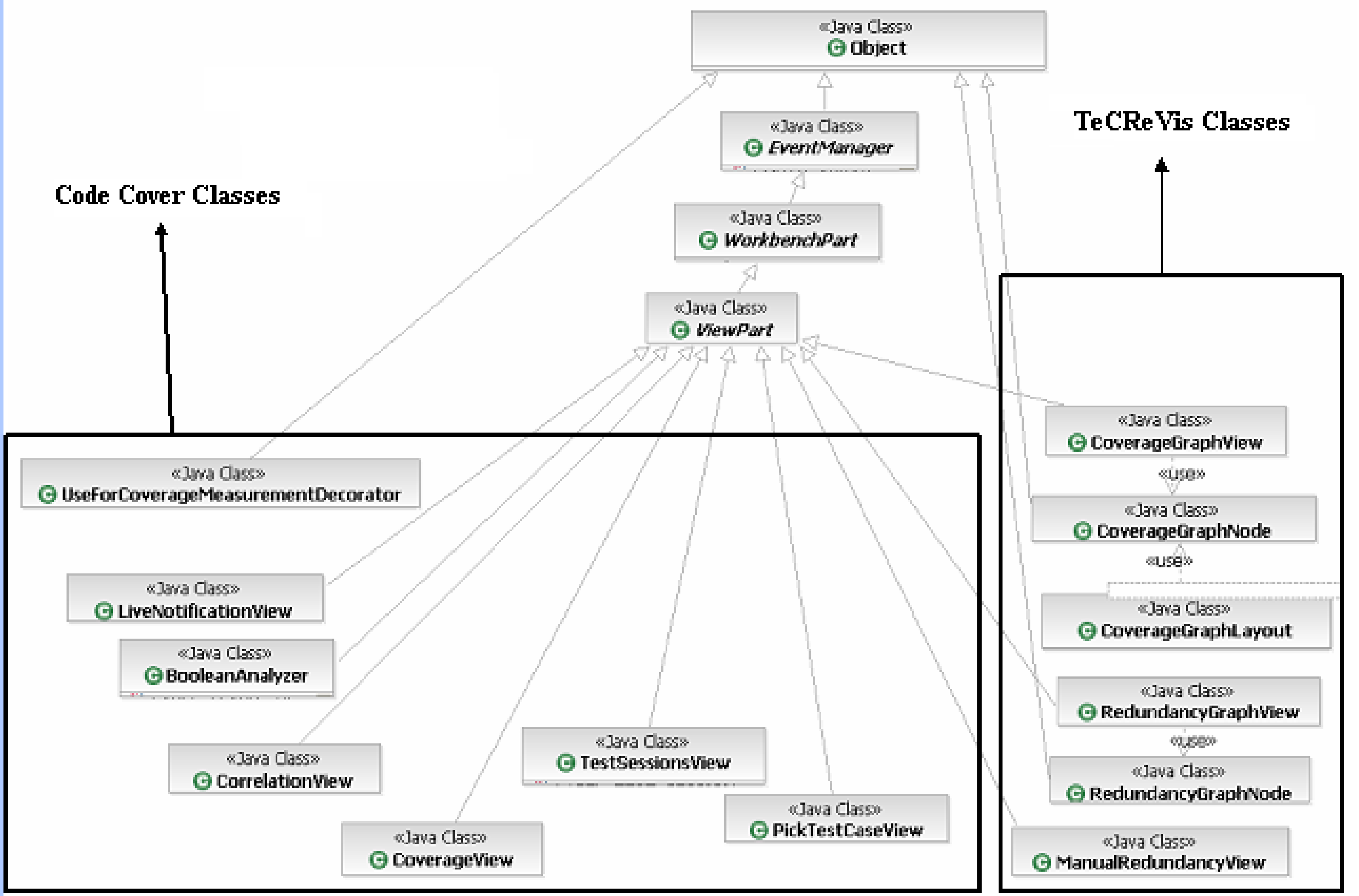

instrumentation
«Java Package» org.codecover.instrumentation
«Java Package» org.codecover.instrumentation.java.measurement
«Java Package» org.codecover.instrumentation.measurement
«Java Package» org.codecover.juniteclipse
report
«Java Package» org.codecover.report
«Java Package» org.codecover.report.csv
«Java Package» org.codecover.report.html
model
«Java Package» org.codecover.model
«Java Package» org.codecover.model.mast
«Java Package» org.codecover.model.utils
metrics
«Java Package» org.codecover.metrics
«Java Package» org.codecover.metrics.correlation
«Java Package» org.codecover.metrics.coverage
eclipse
«Java Package» org.codecover.eclipse
«Java Package» org.codecover.eclipse.annotation
«Java Package» org.codecover.eclipse.tscmanager
«Java Package» org.codecover.eclipse.utils
«Java Package» org.codecover.eclipse.views
«Java Package» org.codecover.eclipse.views.controls
edu.uci.ics.jung

Modified packages in TeReVis

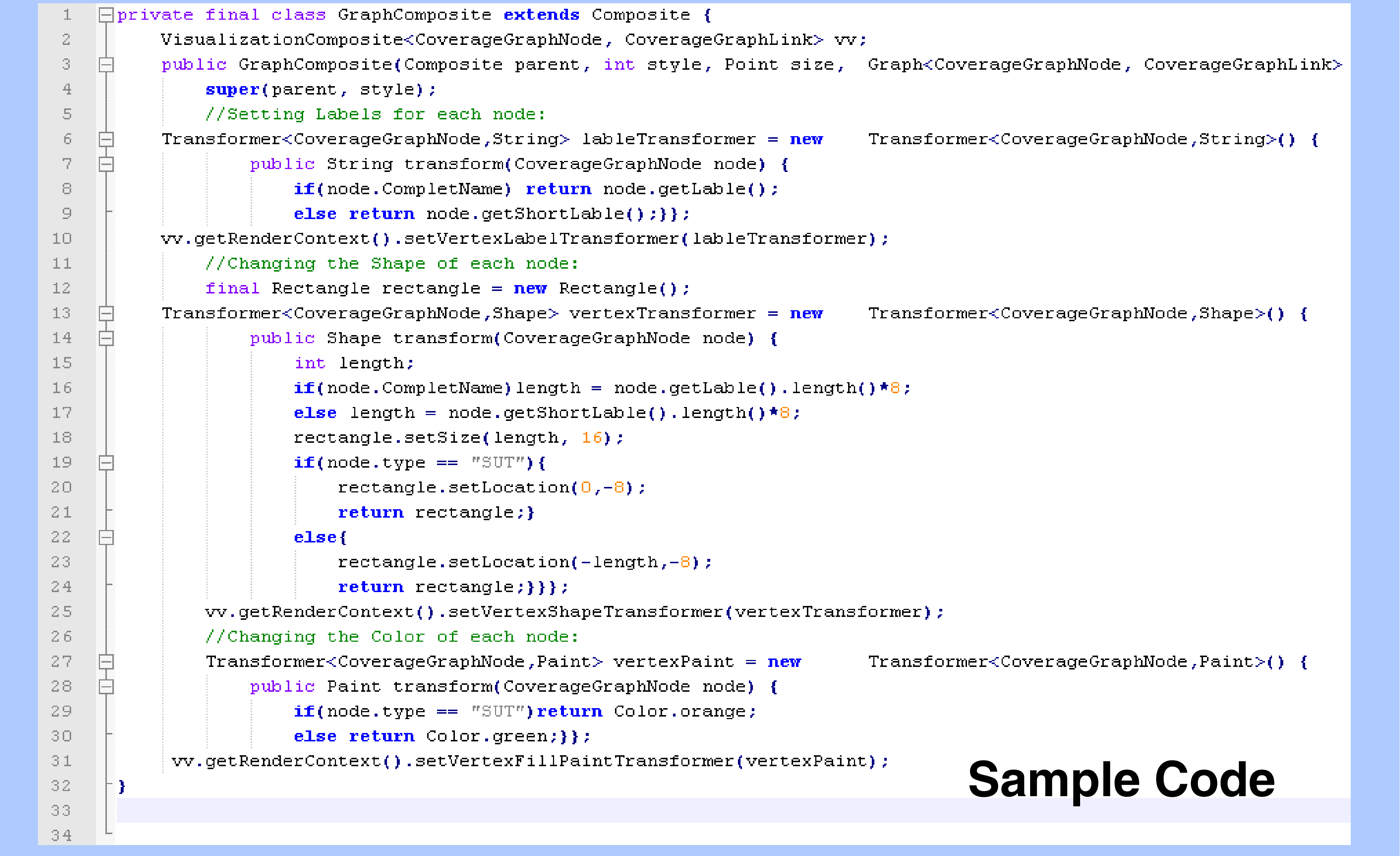
Visualization library Used in TeReVis

TeCReVis - Implementation Details

- **Class Diagram of package `org.codecover.eclipse.views`**



TeCReVis - Implementation Details



```
private final class GraphComposite extends Composite {
    VisualizationComposite<CoverageGraphNode, CoverageGraphLink> vv;
    public GraphComposite(Composite parent, int style, Point size,  Graph<CoverageGraphNode, CoverageGraphLink>
        super(parent, style);
        //Setting Labels for each node:
    Transformer<CoverageGraphNode,String> lableTransformer = new    Transformer<CoverageGraphNode,String>() {
            public String transform(CoverageGraphNode node) {
                if(node.CompletName) return node.getLable();
                else return node.getShortLable();}};
    vv.getRenderContext().setVertexLabelTransformer(lableTransformer);
        //Changing the Shape of each node:
        final Rectangle rectangle = new Rectangle();
    Transformer<CoverageGraphNode,Shape> vertexTransformer = new    Transformer<CoverageGraphNode,Shape>() {
            public Shape transform(CoverageGraphNode node) {
                int length;
                if(node.CompletName)length = node.getLable().length()*8;
                else length = node.getShortLable().length()*8;
                rectangle.setSize(length, 16);
                if(node.type == "SUT"){
                    rectangle.setLocation(0,-8);
                    return rectangle;}
                else{
                    rectangle.setLocation(-length,-8);
                    return rectangle;}}};
        vv.getRenderContext().setVertexShapeTransformer(vertexTransformer);
        //Changing the Color of each node:
        Transformer<CoverageGraphNode,Paint> vertexPaint = new     Transformer<CoverageGraphNode,Paint>() {
            public Paint transform(CoverageGraphNode node) {
                if(node.type == "SUT")return Color.orange;
                else return Color.green;}};
     vv.getRenderContext().setVertexFillPaintTransformer(vertexPaint);
}
```

Sample Code

Talk Outline

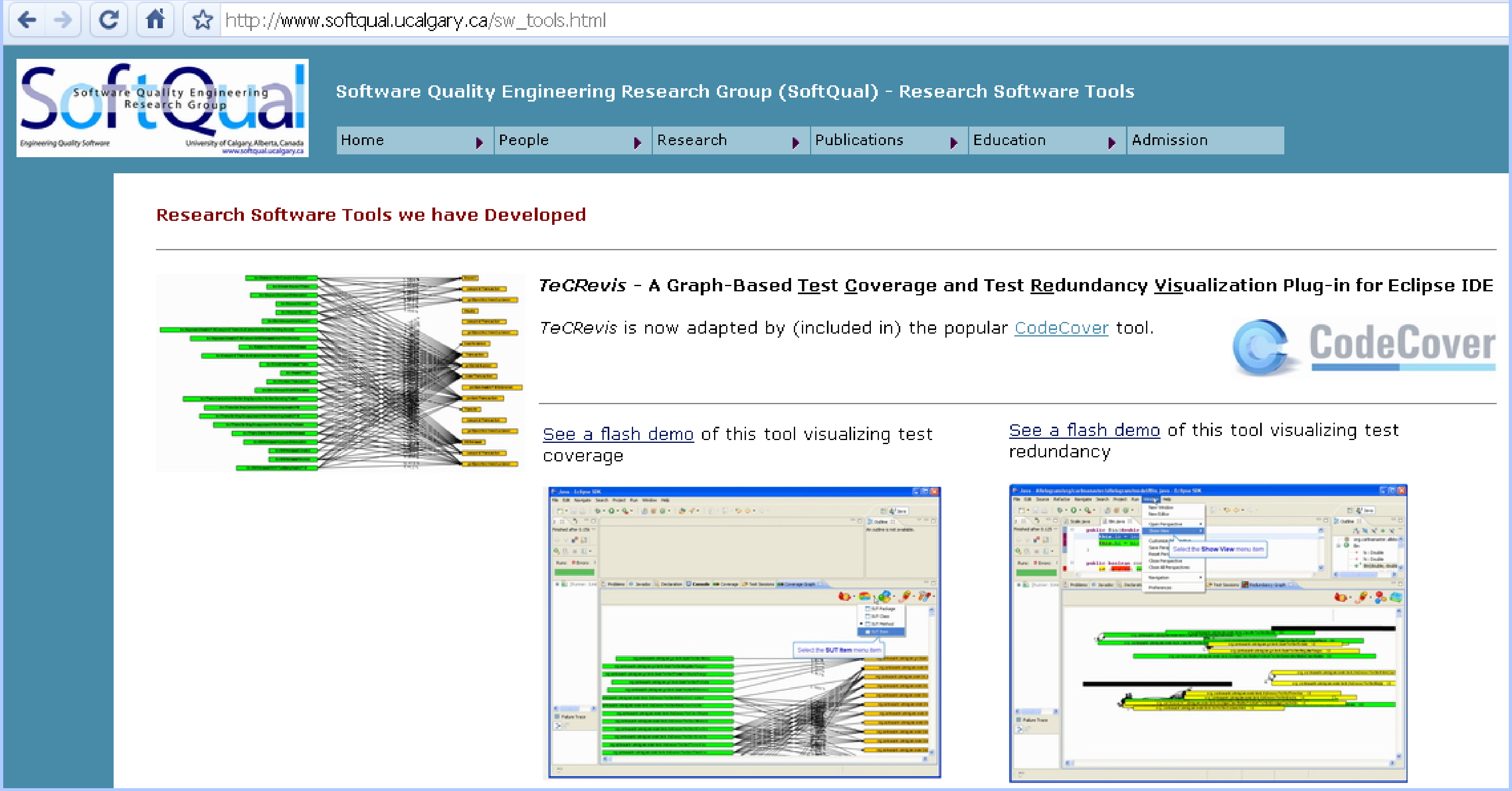
- Existing Code Coverage Tools
- The need for Test Visualization
- 1st Feature of TeCReVis: Code Coverage Visualization
- TeCReVis - Graphical User Interface
- 2nd Feature of TeCReVis: Test Redundancy Management
- TeCReVis - Implementation Details
- **Usage Scenarios of the Tool**
- Availability and Demo Videos
- Q/A

Usage Scenarios of the Tool

- **Coverage (test adequacy) improvement**
- **Test suite maintenance as the SUT evolves**
  - An empirical study is reported in a TAIC PART 2010 fast abstract
- **Fault localization**
  - An empirical study is reported in a TAIC PART 2010 fast abstract
- **Test redundancy detection**
  - For details see…N. Koochakzadeh and V. Garousi “A Tester-Assisted Methodology for Test Redundancy Detection”, Journal on Advances in Software Engineering, Special Issue on Software Test Automation, 2010: pp. 1-13.
  - and N. Koochakzadeh, V. Garousi, and F. Maurer, "Test Redundancy Measurement Based on Coverage Information: Evaluation and Lessons Learned," in *Proc. of Int. Conf. on Soft. Testing, Verification, and Validation (ICST)*, 2009.

Tool availability and Demo Videos…

- **www.softqual.ucalgary.ca/sw_tools.html**



http://www.softqual.ucalgary.ca/sw_tools.html

SoftQual
Software Quality Engineering Research Group
Engineering Quality Software
University of Calgary, Alberta, Canada
www.softqual.ucalgary.ca

Software Quality Engineering Research Group (SoftQual) - Research Software Tools

Home
People
Research
Publications
Education
Admission

Research Software Tools we have Developed

TeCRevis - A Graph-Based Test Coverage and Test Redundancy Visualization Plug-in for Eclipse IDE

TeCRevis is now adapted by (included in) the popular CodeCover tool.

See a flash demo of this tool visualizing test coverage

See a flash demo of this tool visualizing test redundancy

Talk Outline

- Existing Code Coverage Tools
- The need for Test Visualization
- 1st Feature of TeCReVis: Code Coverage Visualization
- TeCReVis - Graphical User Interface
- 2nd Feature of TeCReVis: Test Redundancy Management
- TeCReVis - Implementation Details
- Usage Scenarios of the Tool
- Availability and Demo Videos
- Q/A