A Heuristic Process for GUI Widget Matching Across Application Versions

MaCS IX, February 9-12, Siofok, Hungary
| What’s the problem? |
|---------------------|

GUI applications are ubiquitous

- Lots of tools for design
- Lots of platforms to choose from
- Lots of widget toolkits
- Great testing processes
What’s the problem?

GUI applications are ubiquitous

• Lots of tools for design
• Lots of platforms to choose from
• Lots of widget toolkits
• Great testing processes
Regression testing GUI applications

Reusing GUI test cases

- Using genetic algorithms to repair non-executable tests
  - Si Huang, Myra B. Cohen, and Atif M. Memon. **Repairing GUI test suites using a genetic algorithm.** *Proceedings of the 2010 Third International Conference on Software Testing, Verification and Validation*, pages 245–254, Washington, DC, USA. IEEE Computer Society, 2010

- Inserting and removing events
  - Atif M. Memon. **Automatically repairing event sequence-based GUI test suites for regression testing.** *ACM Transactions Software Engineering Methodology*, 18, 4:1–4:36, November 2008.

- Identifying functionally equivalent GUI elements and updating tests
  - Scott McMaster and Atif M. Memon. **An extensible heuristic-based framework for GUI test case maintenance.** *Proceedings of the IEEE International Conference on Software Testing, Verification, and Validation Workshops*, pages 251–254, Washington DC, USA. IEEE Computer Society, 2009
# Regression testing GUI applications

## Reusing GUI test cases

- Using genetic algorithms to repair non-executable tests
- Inserting and removing events
  - Test cases are altered, new interactions appear
- Identifying functionally equivalent GUI elements and updating tests
  - Exact replay of test cases feasible 😊
- *Only preliminary work available*
Examples of functionally equivalent widgets

FreeMind application - http://freemind.sourceforge.net/wiki/index.php/Main_Page
## GUI Element properties used

| Name   | Description                                                                 |
|--------|-----------------------------------------------------------------------------|
| Accelerator | The shortcut key combination used to trigger the binded action               |
| Class  | The class implementing the widget                                            |
| Icon   | The path to the widget’s icon                                                |
| Text   | The text associated with the widget                                          |
| Title  | The widget’s title.                                                          |
| X      | The X coordinate of the widget’s upper corner in its containing window       |
| Y      | The Y coordinate of the widget’s upper corner in its containing window       |
| Width  | The widget’s width                                                           |
| Height | The widget’s height                                                          |
| Index  | The widget’s index in the parent container                                   |
### Name | Description
--- | ---
Parent | The *Modes* menu
Children | -
Class | `javax.swing.JMenuItem`
Accelerator | Alt+3
Title | -
Icon | -
X | 319
Y | 45
Width | 129
Height | 21
Index | 0

### Name | Description
--- | ---
Parent | The `javax.swing.JToolBar` element
Children | -
Class | `org.gjt.sp.jedit.gui.EnhancedButton`
Accelerator | -
Title | -
Icon | `org/gjt/sp/jedit/icons/Cut24.gif`
X | 196
Y | 47
Width | 30
Height | 30
Index | 7
A heuristic process for widget matching

The process

- Match application windows
  - Window Matching Heuristic
    - Uses window title and root status
- Match window components
  - PropertyValueHierarchyHeuristic
    - Uses property equality, nullity and similarity
    - Works on widget hierarchies
  - PropertyValueHeuristic
    - Uses property equality, nullity and similarity
  - SingletonComponentHeuristic
    - Uses property value uniqueness
  - InverseHierarchyHeuristic
    - Uses container children
- Wrap up the match process
  - Final Heuristic
A heuristic process for widget matching

**FreeMind**
- 13 versions
- November 2000 – September 2007
- Taken from project CVS
- 3500 – 65K lines of code
- Over 13.5 million total downloads

**jEdit**
- 17 versions
- January 2000 – May 2010
- Released versions 2.3pre2 – 4.3.2
- 23K – 106K lines of code
- Over 6.5 million total downloads
## Heuristic metrics

| Using oracle information | Our defined metrics |
|--------------------------|---------------------|
| • Correct Decision Count (CDC) | • Heuristic Correct Decision Count (HCDC) |
|   • The number of correct matching decisions |   • The number of correct decisions taken |
| • Correct Match Count (CMC) | • Heuristic Correct Match Count (HCMC) |
|   • The number of correct matches |   • The number of correct matches performed |
| • Dissimilar Widget Count (DWC) | • Heuristic Correct Decision in Dissimilar Widgets Count (HCDDWC) |
|   • Number of hard-to-match widgets |   • The number of correct decisions in hard-to-match widgets |
Heuristic metrics

- Regardless of GUI size
  
  - Heuristic Decision Rate (HDR). Represents the percentage of correct decisions taken, calculated as $\frac{HDC}{CDC}$.
    
    o *Sensible as a general measure of process accuracy.*
  
  - Heuristic Match Rate (HMR). Represents the percentage of correct matches, calculated as $\frac{HCMC}{CMC}$.
    
    o *Suited for assessing accuracy in test case maintenance.*
  
  - Heuristic Dissimilar Widgets Decision Rate (HDWDR). Represents the percentage of correct decisions taken for dissimilar widgets, calculated as $\frac{HCDDWC}{DWC}$.
    
    o *Shows how accurately the process deals with heavily modified GUI elements*
## Case study results

| Measurement                        | FreeMind | jEdit  | Total  |
|-----------------------------------|----------|--------|--------|
| Correct Decision Count            | 1799     | 8976   | 10775  |
| Correct Match Count               | 1524     | 7461   | 8985   |
| Dissimilar Widget Count           | 797      | 4115   | 4912   |
| Heuristic Correct Decision Count  | 1743     | 8436   | 10179  |
| Heuristic Correct Match Count     | 1502     | 7194   | 8696   |
| Heuristic Decision Rate           | 96.89%   | 93.98% | 94.47% |
| Heuristic Match Rate              | 98.56%   | 96.42% | 96.78% |
| Heuristic Dissimilar Widget       | 89.46%   | 80.46% | 81.92% |
Result consistency

FreeMind

![Graph showing result consistency over time with different categories: HDR, HMR, HDWDR.](image-url)
Result consistency

jEdit

![Graph showing result consistency for HDR, HMR, and HDWDR with various percentages for different versions. The graph compares the performance of HDR, HMR, and HDWDR across different files and versions, indicating a trend in their consistency.](image-url)
Heuristic error analysis

Multiple changes to widgets

Changing widget types

Changes beyond GUI layer

Changes in complex widgets
Conclusions & Future work

- Toolset and case study repository home
  - [https://sourceforge.net/projects/javaset/](https://sourceforge.net/projects/javaset/)
- Improve the matching process
  - Record complex widget data models
  - Implement custom widget comparators
- Extend case study with .NET, SWT applications
Thank You!