Scriptless test automation through graphical user interface (GUI)

Pekka Aho, Open Universiteit
The outrageous promise

Script less and test more!
Scripted test automation through GUI

- Test steps defined prior to execution
- Each check separately defined
- Maintenance when the GUI changes
Scripted test automation through GUI

Captured Selenium script

- Easy to capture
- Readable?
- Maintenance?

<table>
<thead>
<tr>
<th>Command</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>open</td>
<td><a href="http://www.testdag.nl/">http://www.testdag.nl/</a></td>
</tr>
<tr>
<td>set window size</td>
<td>738x803</td>
</tr>
<tr>
<td>select frame</td>
<td>index=0</td>
</tr>
<tr>
<td>click</td>
<td>css=.material-icons</td>
</tr>
<tr>
<td>click</td>
<td>linkText=Location</td>
</tr>
<tr>
<td>click</td>
<td>linkText=Capgemini headquarters in Leidsche Rijn (Utrecht)</td>
</tr>
</tbody>
</table>
Scripted test automation through GUI
Scripted test automation through GUI
Scriptless test automation through GUI

- Tests are generated during the execution

1. Detect all the available interactions
2. Select and execute one interaction
3. Wait for the GUI to update, check test oracles
Scriptless test automation through GUI

- Usually some randomness involved
- Defining test oracles is different
Open source TESTAR tool

Tool for scriptless GUI test automation
- [https://testar.org/](https://testar.org/)
- [https://github.com/TESTARtool/TESTAR_dev](https://github.com/TESTARtool/TESTAR_dev)

Active research and feature development
- Machine learning and artificial intelligence
- Model inference and automated change detection
Open source TESTAR tool
Obtaining the state of the GUI

• Using Windows Accessibility API or Selenium Web Driver

• Obtaining a widget tree
Obtaining the state of the GUI

Click on the "volume off" button.

Click on the "volume on" button.
Deriving actions

Configuring the types of actions

• Left click on menus and buttons
• Left click + type text on input fields
• Sliding actions for scrollbars
• Drag and drop?
• Double clicks?
• Right clicks?
Filtering actions

Configuring the actions NOT to select

- Text based filtering: "Exit", "Close", "Minimize", ...
- Specific widget filtering
- Programmatic filtering
Selecting an action

- Using an action selection algorithm
- In a specific state, triggering predefined behaviour
Action selection algorithms

- Random
- Random + prioritize new actions
- Learning a model, exploring new actions
- Machine learning
- Artificial intelligence
- Evolutionary algorithms
Test oracles

- Generic "free" oracles: crash, freeze
- Suspicious titles: "Error", "Exception", ...
- Oracles that trigger with a matching GUI state

1. Detect all the available interactions
2. Select and execute one interaction
3. Wait for the GUI to update, check test oracles

Crashed? Freezed?
Output of TESTAR tool

• Executable test sequence that leads to failure
• HTML report with screenshots
• GUI state graph database (if enabled)
Scripted vs. Scriptless GUI Test Automation

**Scripted:**
- Specific paths (user scenarios)
- Precise test oracles
- Manual effort to create and maintain the test scripts

**Scriptless:**
- Test generation
- Low maintenance effort
- More challenging test oracles
- Requires execution time to get good coverage

Scripted smoke tests and critical test scenarios

---

Scriptless nightly testing for robustness and coverage
Any questions?

More information:

• [https://testar.org/](https://testar.org/)
• Email: [pekka.aho@ou.nl](mailto:pekka.aho@ou.nl)
• [https://www.testomatproject.eu/](https://www.testomatproject.eu/)