Keynote: Trends and challenges in automated GUI testing

The 1st International Workshop on User Interface Test Automation
19 October 2015, Sophia Antipolis, France

Pekka Aho, VTT, Finland

(Tanja Vos), Urko Rueda Molina, UPV, Spain

Why GUI test automation is important?

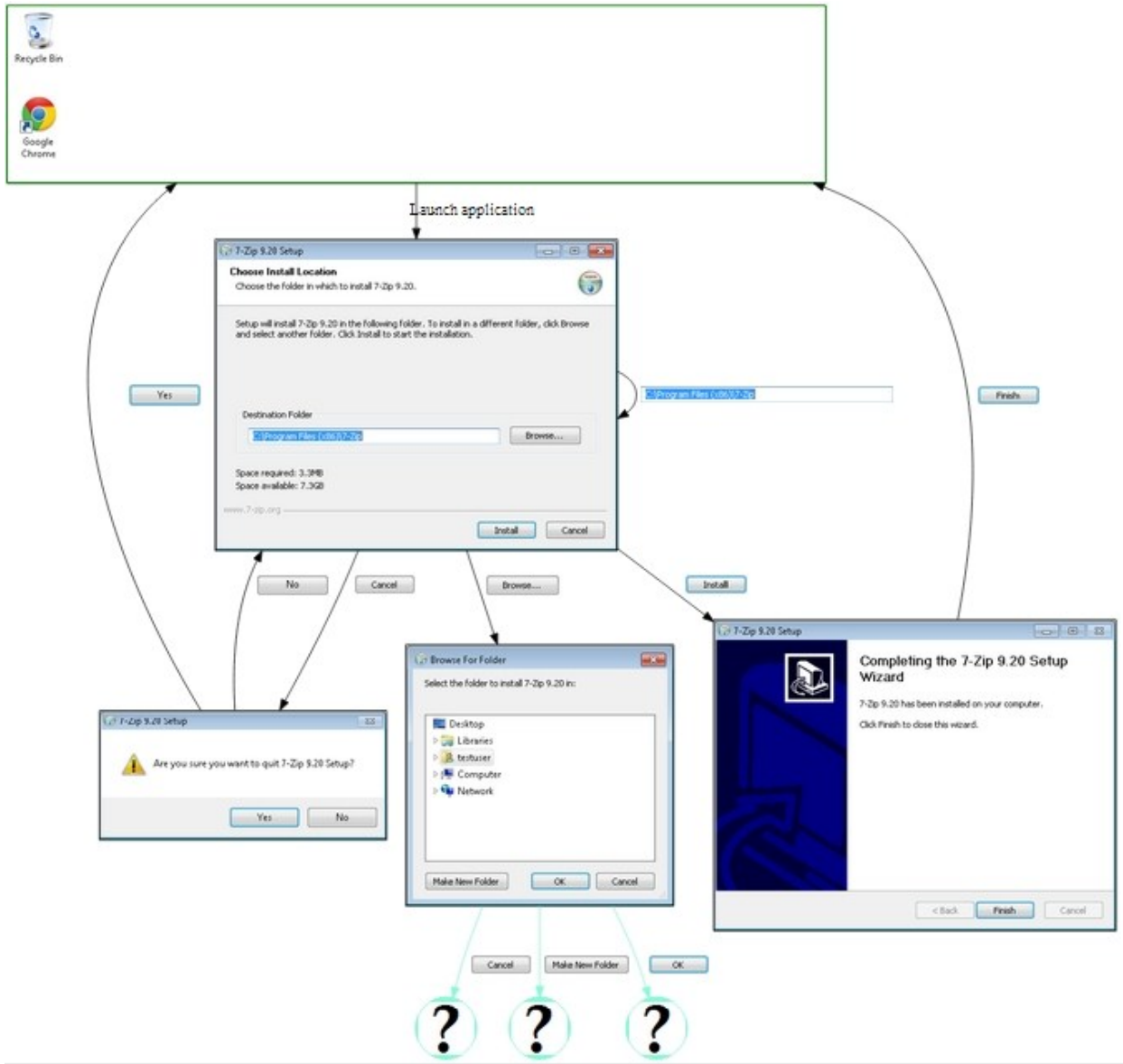
- As for a large part of software testing, agile and continuous integration processes and short development cycles set time limits also for UI testing, and the answer is automation
- Most people are more or less dependent on their mobile phones and all kinds of mobile and web apps, most having a GUI

Trends in automated GUI testing research

- Today presented in INTUITEST:
  - Automated debugging of errors found during automated GUI testing
  - Creating executable tests for different versions (on different platforms) of GUI for the same application from a single abstract description
  - Surveying the level and issues hindering the adoption of automated GUI testing (in Brazil)
  - Combining various GUI testing approaches (e.g., capture&replay and model-based testing) into a tool chain
  - Using property-based testing to automate web GUI testing
- Although not presented as a main topic today, model extraction using dynamic GUI crawling is a popular research topic

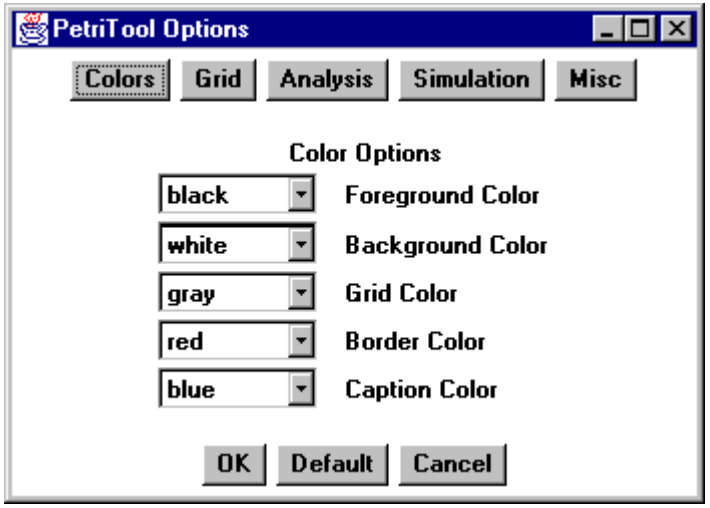
GUI model extraction using dynamic analysis

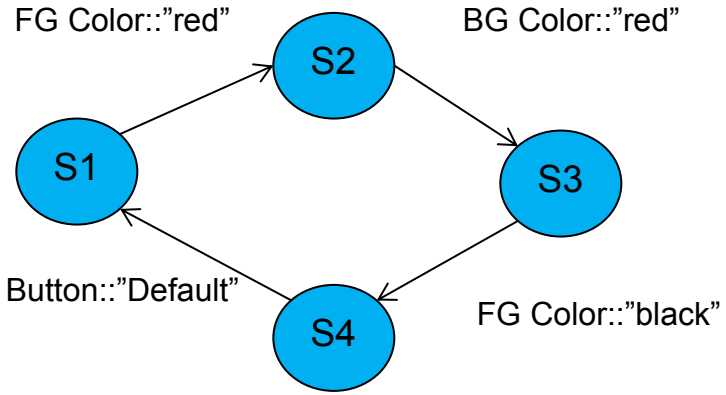
1. Starting the GUI application through the model extraction tool
2. Letting the tool automatically explore the GUI and create a model
   I. Analyzing the state of the GUI and actions that are available
   II. Selecting the action (potential state transition) to execute
   III. Executing the action and waiting for the GUI to update
3. Depending on the tool and the GUI app, some manual effort is usually required, for example:
   - Providing valid input for screens such as login with password
   - Providing instructions for the tool if the automation is missing something (and restart the modeling with the new instructions)
   - Possibly several iterations of trying if the instructions are sufficient, and maybe manually validating the extracted model (bottom-up)

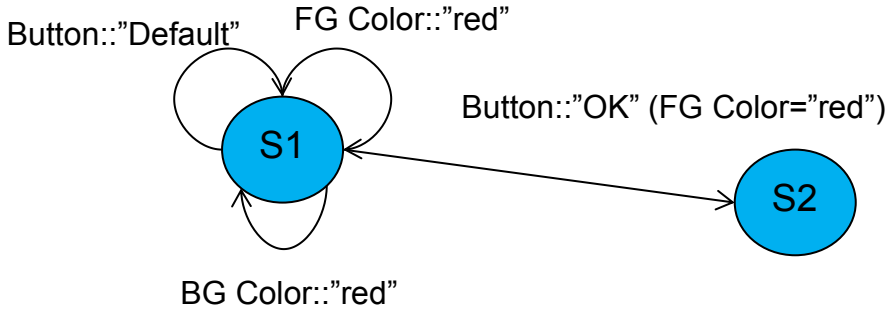

Recycle Bin
Google Chrome
Launch application
7-Zip 9.20 Setup
Choose Install Location
Choose the folder in which to install 7-Zip 9.20.
Setup will install 7-Zip 9.20 in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.
Destination Folder
Browse...
Space required: 3.3MB
Space available: 7.3GB
Install
Cancel
Yes
No
Finish
Browse For Folder
Select the folder to install 7-Zip 9.20 in:
Desktop
Libraries
testuser
Computer
Network
Make New Folder
OK
Are you sure you want to quit 7-Zip 9.20 Setup?
Completing the 7-Zip 9.20 Setup Wizard
7-Zip 9.20 has been installed on your computer.
Click Finish to close this wizard.
< Back
?

What to do with the extracted GUI models?

- Challenges: no link to the requirements, the level of abstraction vs. state space explosion, reaching all the parts of the GUI in a reasonable time





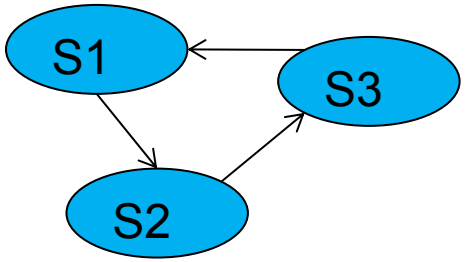


What to do with the extracted GUI models?

- Challenges: no link to the requirements, the level of abstraction vs. state space explosion, reaching all the parts of the GUI in a reasonable time
- The extracted GUI models can be used for example:
  - to automate testing (easy mapping to the implementation)
  - for program analysis and understanding
  - for migrating legacy applications to new technologies
  - for documenting an existing application
- A lot of “testing” happens already during the automated model extraction
- Most approaches use the extracted GUI models to generate test sequences and execute them against the consequent versions of the same GUI for reference testing

Using the extracted models to generate tests

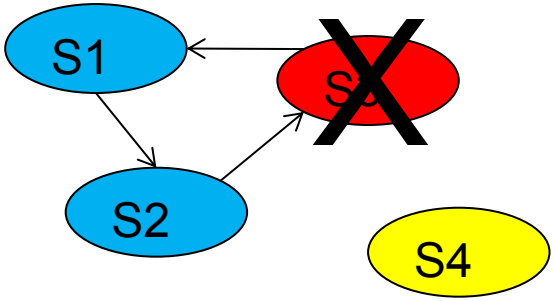
1. Extract the model from version 1.0



2. Generate test sequences from the model

3. Execute the tests to get the oracle (reference)

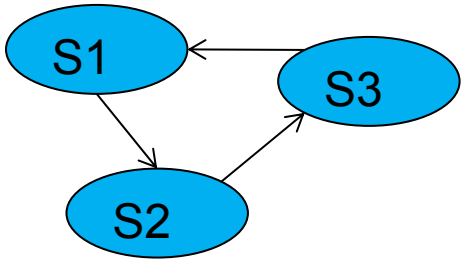
4. Execute the tests on version 2.0



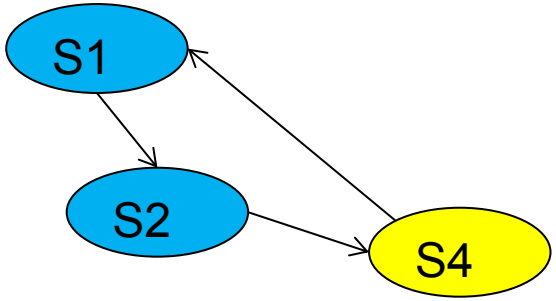
Tests on the removed parts are failing and the new parts are missing from the tests
- Extract a new model, then the reference is missing

Using the extracted models to find the changes

1. Extract the model from version 1.0



2. Extract the model from version 2.0



3. Compare the models to find the changes

Murphy tool: https://github.com/F-Secure/murphy

More trends in automated GUI testing research

- Combining static analysis to the dynamic model extraction
  - Extract better and more detailed models, generate better tests from the extracted models, help in debugging or finding the errors
- Automated reference testing (finding the changes compared to a previous version) research and tools are becoming mature enough for industrial use, but not yet widely used

Challenges in automated GUI testing

- What to automate and how to get the intelligence from the user (or the specs) – in which phase the manual effort is required
  - Capture&Replay and scripts – automating the test execution
    - C&R easy and intuitive to use but lot of maintenance effort
    - Scripting requires more skill but might be easier to maintain
  - Model-based testing – automating also the test case generation
    - Requires expertise in modeling and tooling, mapping to the implementation for executable tests
  - Model extraction – automating also the modeling and mapping
    - Not automated: link to the requirements, test data
  - Measurement, visualization, reporting, debugging, optimization, etc

Challenges in automated GUI testing

- Lack of platform independent means for GUI automation
  - Usually a separate implementation is required for each platform
  - Image recognition from screenshots is independent but inaccurate
- Widgets requiring complex input (e.g., drawing tool, drag&drop)
  - Difficult for the GUI automation tool to simulate a real user - usually manually given specific instructions required
- Capturing the context of GUI actions
  - Basically all the previous actions from the start (or even installation) of the application might affect its behavior
- New GUI technologies
  - Touch screens
  - Multitude of different kinds of HW and SW platforms
  - Dynamically (during run-time) created web GUIs

What is hindering the industrial adoption?

- Manual GUI testing and “old school” capture&replay tools are still widely used
- State-of-the-art automated GUI testing approaches and tools are not widely adopted into use in practice (industry)
  - Tool support (installation and test environment, usability, learning curve, ...)
  - Lack of knowledge on the existence and benefits of the latest approaches and tools

Thank you! Any questions?

For further information contact: Pekka.Aho@vtt.fi