



**Open Universiteit**



**Universiteit  
Utrecht**



# Large Language Models in the scriptless test loop

**Paul Tieleman**

Master student SE at the OU

Internship at ING





The app gets bigger and more complex, which means exponentially more testing effort...





All these testing  
scripts need to be  
updated  
constantly...









This testing is  
costing me so  
much time...







I'm always afraid that  
I'm missing an edge  
case somewhere...



# The problem

- More complex software -> more testing effort
- Testing effort:
  - Updates
  - New scripts
  - Uncertainty
- More effort -> More test engineers -> More costs
- Higher expectations

# AUTOLINK project

- Automated Unobtrusive Techniques for LINKing requirements and testing in agile software development
- Goal: Automate end-to-end testing based on requirements
- Scriptless testing



Open Universiteit



Universiteit  
Utrecht



axini

newspark  
a BlueGroup IT company



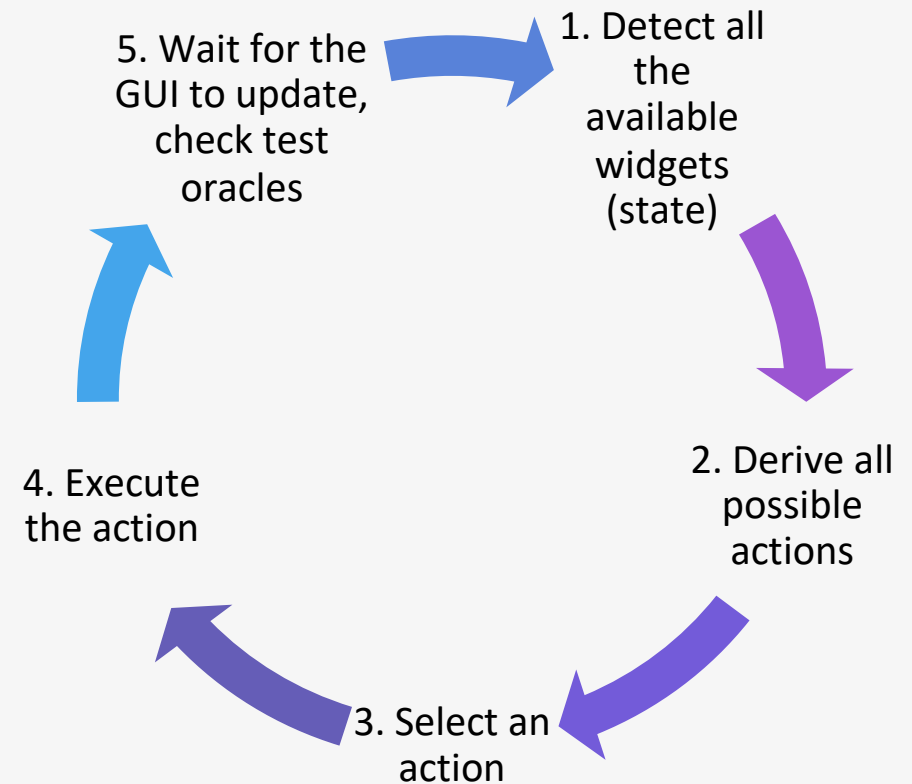
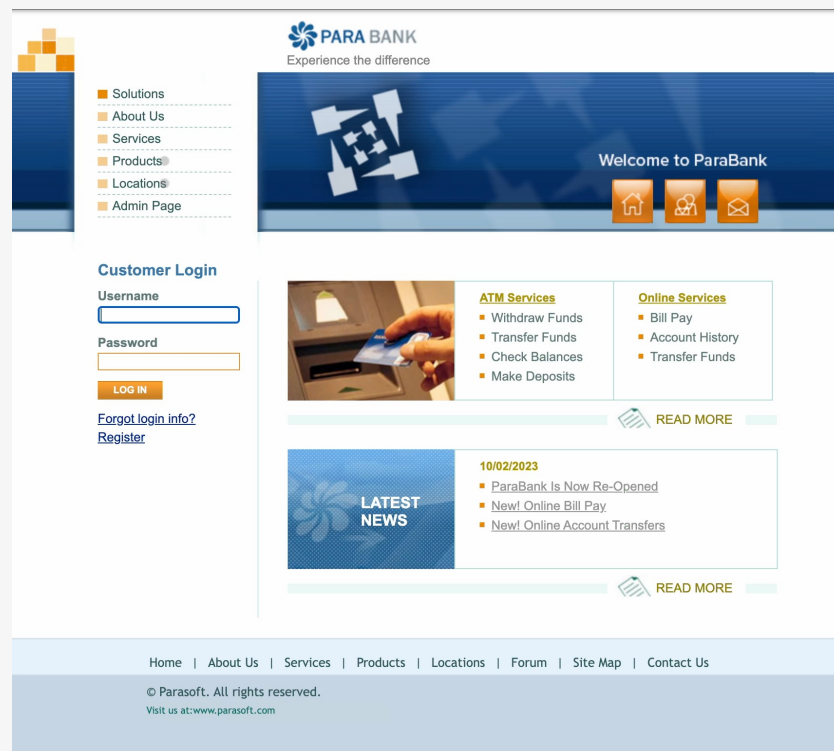
mendix

TESTCompass



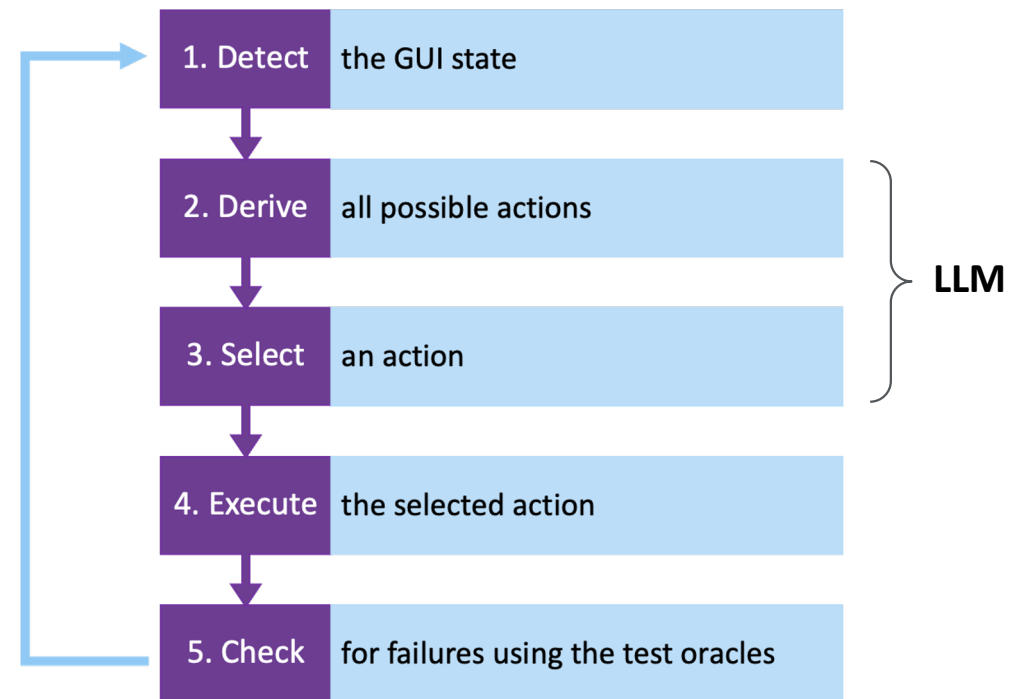
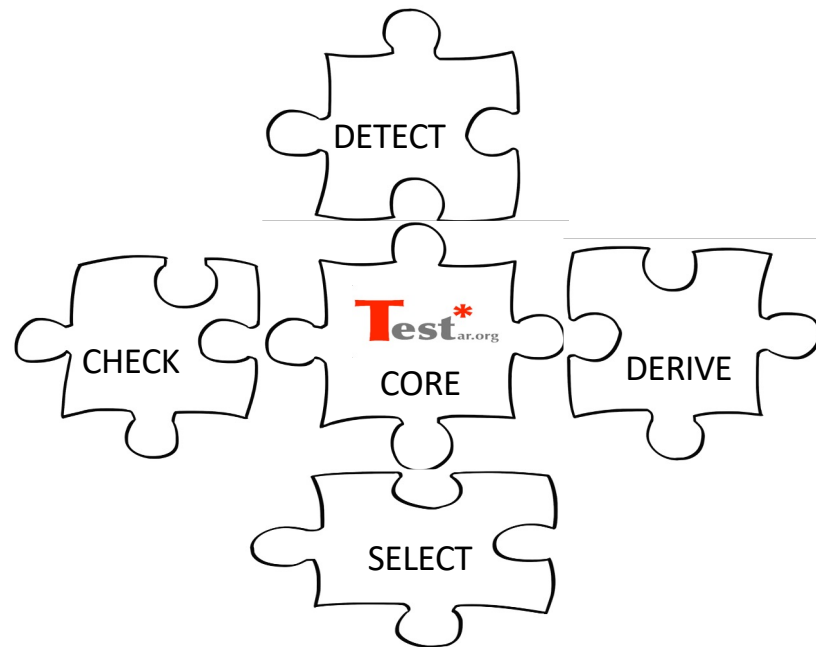
# Background

- Scripted testing -> high maintenance
- Scriptless testing tool: Testar



# WHAT

- Special focus on action derivation & selection
- User interaction by chat
- Use case test script generation



# Sneak preview

```
>>> List all actionable HTML elements that a user can interact with in the following HTML code: <html lang="en" data-framework="typescript" id="0"><head id="0.0"><meta charset="utf-8" id="0.0.0"><title id="0.0.1">React • TodoMVC</title><link rel="stylesheet" href="node_modules/todomvc-common/base.css" id="0.0.2"><link rel="stylesheet" href="node_modules/todomvc-app-css/index.css" id="0.0.3"><style id="0.0.4"> label { line-height: 2em; margin-bottom: 0.75em; } input { height: 2em; margin-left: 0.75em; } </style><script src="chrome-extension://jelmbjplhknngggaigfkbiceoaoniniig/js/testar.state.js" id="0.0.5"></script><script src="chrome-extension://jelmbjplhknngggaigfkbiceoaoniniig/js/testar.canvas.js" id="0.0.6"></script></head><body id="0.1"><section class="react-app" id="0.1.0"></section><section class="todoapp" id="0.1.1"><div id="0.1.1.0"><header class="header" id="0.1.1.0.0"><h1 id="0.1.1.0.0">todos</h1><input class="new-todo" placeholder="What needs to be done?" id="0.1.1.0.0.1"></header></div></section><section class="controls" id="0.1.2"></section><section class="flow" id="0.1.3"></section><footer style="font-size: 1em;" class="info" id="0.1.4"><p id="0.1.4.0">Double-click to edit a todo</p><p id="0.1.4.1"><a href="#" id="0.1.4.1.0">The todo app</a></p><p id="0.1.4.2"><a href="#" id="0.1.4.2.0">The controls app</a></p><p id="0.1.4.3"><a href="#" id="0.1.4.3.0">The flow app</a></p></footer><script type="text/javascript" src="node_modules/director/build/director.js" id="0.1.5"></script><script type="text/javascript" src="js/bundle.js" id="0.1.6"></script><canvas id="0.1.7" width="1728" height="865" style="left: 0px; top: 0px;"></canvas></body></html>
```

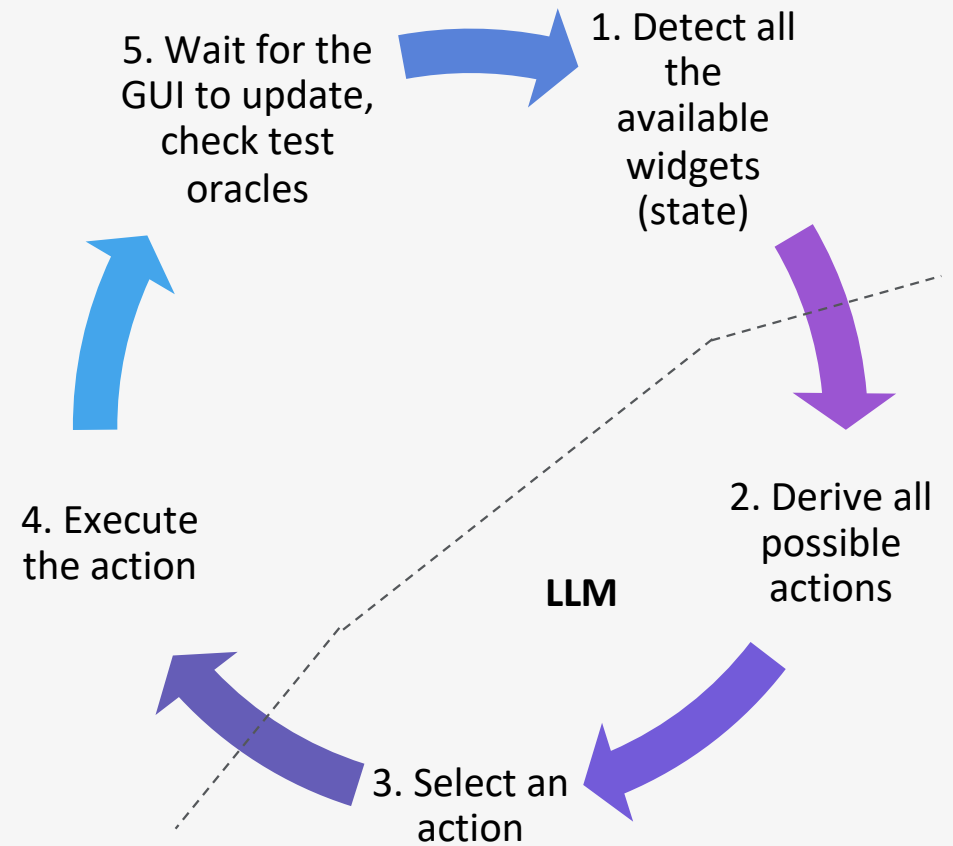
# WHY

- Constantly evolving GUI leads to high maintenance test scripts
- Merging of use cases and test scripts
- Why LLMs?
  - HTML code -> action derivation
  - Large body of text -> use case comprehension, test script generation, and action selection
- 3 main use cases with Testar -> Improvement of interaction with Testar



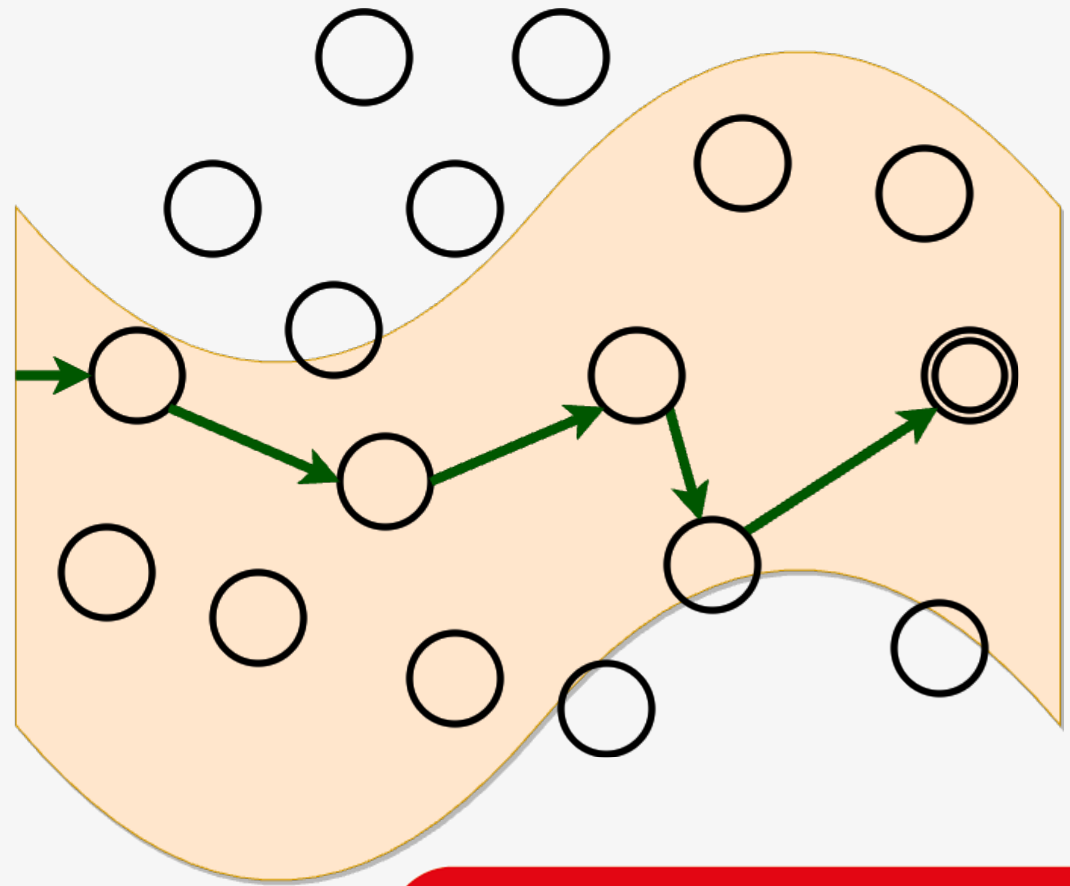
# HOW

- LLM in the Testar loop
- Chat plugin
- Use case test script generation
- Combination with default prompts



# Future work

- Listener mode
- Fine tuning of test trace bandwidth
- LLM oracle
- Let LLM identify webpage for better context





**Paul Tieleman**

Student Master SE at the OU

Internship at ING

