RUHR UNIVERSITÄT BOCHUM RUB

Bachelor Topic

Integrating Script-free Testing with the Playwright Framework

Motivation and Background:
Traditional web GUI testing requires significant maintenance effort due to its heavy reliance on fragile locators to interact with elements, which often break as applications evolve [1]. To abstract element localization, recent research introduced script-free testing [2, 3], where test cases are written in a natural or domain-specific language and automatically interpreted without manual scripts. Kirinuki et al. [2, 3] developed a corresponding tool to demonstrate script-free testing; however, experiments revealed some limitations related to the *waiting* problem. This is the uncertainty of when a web element will appear on the page and become ready for interaction. Thus, even when the tool identifies the correct element, it may fail to interact with it at the right time during execution. Their current tool relies on Selenium to interact with the web page, where synchronization issues are a persistent challenge [4]. Selenium lacks automatic waiting and reliable handling of asynchronous operations, leading to unpredictable behavior during tests. However, modern frameworks like Playwright include robust waiting mechanisms capable of addressing this challenge [5, 6]. This thesis proposes enhancing the script-free tool by integrating Playwright and evaluating the improvements through reproducible experiments, aiming to provide a more robust solution for testing dynamic web applications.

Student Task and Responsibilities:
- Replace the current Selenium-based execution layer of the script-free testing tool with Playwright.
- Leverage Playwright's auto-waiting and network synchronization features to improve test execution reliability.
- Design and conduct reproducible experiments on realistic web applications to assess improvements in test stability and element interaction success.
- Analyze the results and compare them with the Selenium-based tooling, in particular, with respect to the observed synchronization issues.

Deliverables:
- A script-free testing tool using the Playwright framework.
- Comprehensive experiments using existing and new benchmarks demonstrating the impact of the integration.
- Documentation of the implementation process, experimental setup, and evaluation findings.

Pre-Requisites: (Programming Languages, OS, Skills, Papers, etc)
- Experience with JavaScript or TypeScript and familiarity with web testing frameworks such as Selenium or Playwright.
- Basic understanding of web technologies (DOM, asynchronous behavior, dynamic content).
- Ability to read and modify academic tools or prototypes.
- Read the script-free related works [2, 3], Playwright documentation, and testing concepts (auto-waiting, selectors, browser contexts).

[1] Leotta, Maurizio, et al. "Challenges of end-to-end testing with selenium WebDriver and how to face them: A survey." *2023 IEEE Conference on Software Testing, Verification and Validation (ICST)*. IEEE, 2023. https://doi.org/10.1109/ICST57152.2023.00039

[2] H. Kirinuki, S. Matsumoto, Y. Higo and S. Kusumoto, "NLP-assisted Web Element Identification Toward Script-free Testing," 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME), Luxembourg, 2021, pp. 639-643, doi: 10.1109/ICSME52107.2021.00072.

[3] H. Kirinuki, S. Matsumoto, Y. Higo and S. Kusumoto, "Web Element Identification by Combining NLP and Heuristic Search for Web Testing," 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), Honolulu, HI, USA, 2022, pp. 1055-1065, doi: 10.1109/SANER53432.2022.00123.

[4] Nass, Michel, Emil Alégroth, and Robert Feldt. "Why many challenges with GUI test automation (will) remain." *Information and Software Technology* 138 (2021): 106625. https://doi.org/10.1016/j.infsof.2021.106625

[5] García, Boni, et al. "Exploring Browser Automation: A Comparative Study of Selenium, Cypress, Puppeteer, and Playwright." *International Conference on the Quality of Information and Communications Technology*. Cham: Springer Nature Switzerland, 2024. https://doi.org/10.1007/978-3-031-70245-7_10

[6] Microsoft. (n.d.). *Auto-waiting*. Playwright. Retrieved April 29, 2025, from https://playwright.dev/docs/actionability

Contacts

Thiago Santos de Moura (thiago.santosdemoura@ruhr-uni-bochum.de)
Software Quality Group, Faculty of Computer Science, Ruhr University of Bochum