

## Bachelor Topic

# Leverage LLM for Maintenance of Web Element Locators in Automated GUI Testing

### Motivation and Background:

To automate web GUI testing, DOM-based GUI test scripts can be used. They heavily rely on locators to interact with elements on a web page, such as identifying and filling input portions (e.g., form fields), executing computations by locating and clicking buttons, and verifying output correctness by locating result-displaying elements [1]. Locators must be regularly checked for accuracy and may need updates at each software release. Sometimes, even a slight modification of the SUT has a massive impact on locators, causing test breakages. These test breakages are not defects or bugs in the SUT but failures in the test automation code that require additional maintenance. The time required for this maintenance can range from quick and simple fixes to slow and complex ones, but in all cases, it takes time away from developing new features [2].

LLMs can reduce this maintenance effort by recovering broken locators or selecting new ones based on similarity with the intended element [3]. Although the original VON Similo study used GPT-4, it raised important concerns regarding cost and privacy [4]. Since then, the availability of open-source and locally deployable models has increased, making it feasible to explore solutions that run without external API dependencies. Moreover, the study did not apply prompt engineering, which opens opportunities to assess whether structured techniques such as Chain-of-Thought can improve the precision of element selection. Recent works have also been limited to Selenium and Robot Framework [4, 5]. At the same time, modern frameworks like Cypress and Playwright are widely used in web development and have different interaction patterns and locator behaviors. These characteristics may influence the effectiveness of LLM-based assistance and deserve further investigation.

### Student Task and Responsibilities:

- Implement the state-of-the-art approach in a modern web testing framework like Cypress or Playwright.
- Propose and include improvements on how locators are fixed or how dynamic locators can be used to be resilient to changes in the SUT.
- Run experiments using local and open-source LLMs to compare the effectiveness of each approach.
- Investigate the application of prompt engineering techniques in the context of locator repair.
- Analyze the results and document your findings.

### Deliverables:

- An implementation of a tool or dependency that automatically fixes locators in Cypress or Playwright scripts using LLMs.
- Evaluation artifacts, such as datasets, tools, and benchmark results.
- Documented findings of the conducted experiments.

### Pre-Requisites: (Programming Languages, OS, Skills, Papers, etc)

Experience with JavaScript or TypeScript, especially with web testing frameworks. Familiarity with LLMs and prompt engineering techniques is also recommended.

[1] Moura, Thiago Santos de, et al. "An Automatic Approach for Uniquely Discovering Actionable Elements for Systematic GUI Testing in Web Applications." *2024 IEEE 24th International Conference*

on *Software Quality, Reliability and Security (QRS)*. IEEE, 2024. <https://ieeexplore.ieee.org/abstract/document/10684641>

[2] Nass, Michel, et al. "Robust web element identification for evolving applications by considering visual overlaps." *2023 IEEE Conference on Software Testing, Verification and Validation (ICST)*. IEEE, 2023. <https://ieeexplore.ieee.org/abstract/document/10132199>

[3] Xu, Zhuolin, Qiushi Li, and Shin Hwei Tan. "Guiding ChatGPT to fix web ui tests via explanation-consistency checking." *arXiv preprint* (2023). <https://arxiv.org/abs/2312.05778>

[4] Nass, Michel, Emil Alégroth, and Robert Feldt. "Improving web element localization by using a large language model." *Software Testing, Verification and Reliability* 34.7 (2024): e1893. <https://onlinelibrary.wiley.com/doi/full/10.1002/stvr.1893>

[4] Nass, Michel, Emil Alégroth, and Robert Feldt. "Improving web element localization by using a large language model." *Software Testing, Verification and Reliability* 34.7 (2024): e1893. <https://onlinelibrary.wiley.com/doi/full/10.1002/stvr.1893>

[5] Rohamo, Paavo. "Enabling Self-healing Locators for Robot Framework with Large Language Models." Master Thesis (2024). <https://helda.helsinki.fi/bitstreams/631b961a-8642-42ed-9826-3e196eac9cf7/download>

## Contacts

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