

WEB BREAKAGES ROOT CAUSE ANALYSIS AND AUTOMATIC REPAIR USING COMPUTER VISION

Motivation:

Investigate how to combine the benefits of DOM-based and visual technology in the same test automation tool. The approaches are complementary, e.g., DOM-based *asserts on the model* of the page (DOM), visual *asserts on the appearance* of the page (GUI), but neither of them can do both.

Context:

Studying in which contexts the (DOM+Visual) approach could be more beneficial, for instance (1) root cause analysis and (2) automatic repair of test suites.

Background:

Automatic repair of broken test suite code is a research area which is far from being consolidated. Breakages may for instance involve (1) structure (e.g., non selection & mis selection of locators), (2) content, (3) workflow. Moreover,

The intuition is that different breakages might need different repair strategies. Often repairing requires inspecting the SUT or analyzing the test as it executes (including its visual execution trace) to understand *what* kind of repair is needed and *where* (JUnit error message are not always useful and can be deceiving for some types of error). Computer vision might have a role to provide a better understanding, and a more accurate repair technique.

Related Work (closest):

- ReASSERT (JUnit): repair assertions and incorrect portions of test code with the correct values derived from the test execution. Limitations: repair only assertions (i.e., direct breakages).
- WATER: repair locators at the breakage site (i.e., direct breakages). Uses a DOM-based differencing technique to find the element on the new DOM. Limitations: (1) the tool might not find a repair. Indeed, it is ineffective when target elements have different tag name (e.g., input -> button), which is rare but can happen, (2) the tool might find many false positives due to heuristic similarity index, (3) limited to locators only, (4) might be deceiving in case of propagated breakages, and (5) inapplicable in case of silent breakages. WATERFALL uses WATER repair strategy but it does not add anything. MULTILOCATOR targets robustness and is theoretically useful for repair, but targets locator and direct breakages only.