Using CrawlJax to Evaluate Website User Experience

Peter Ballen, Kevin Roberts

October 2013

Abstract

With the web becoming a larger part of more and more peoples' lives and with most companies having some form of website, the need for a simple way to assess usability is paramount. Most web developers have too much experience to be objective, and performing usability testing can be a long and time consuming process. We will write a plugin for the dynamic web crawler CrawlJax, which examines state changes by examining the DOM tree of webpages and then constructs a visual state-flow graph of the connections. By writing a plugin to approximate human behavior at various points in the crawling process, we hope to create a tool which can create benchmarks of how usable a website is and create visual abstractions of how a human is likely to explore a website.

1 Introduction

Humans tend to browse websites in a fairly standard way: they start on the homepage and click links to browse through the pages. Pages that require following a single well-labeled link are easily accessible from the homepage. Pages that require following several links are far less accessible. Additionally, a page may not be accessible even if it one link away from the homepage, depending on the visibility of that link

We will be creating a tool to estimate the accessibility of a webpage from the homepage. If an important page has a low accessibility rating, it may be wise to redesign the architecture of the webpage.

2 Previous Work

CrawlJax is a tool designed to dynamically look through AJAX webpages. [1] By looking at the Document Object Model (DOM) of a webpage, it can look for user interactions which could cause a change in the DOM tree. If a new state is different enough, it is added to a state-flow graph. This API has also been used for determining accessibility of AJAX applications [2] as well as for clustering websites based on certain features [3].

Other tools exist for discovering usability issues in websites. The Bloodhound Project [4] uses Infoscent Simulations to approximate how users navigate along a website using ”proximal cues” such as font size and color and other information they already know or learned about the goal. This technique resulted in moderate to strong correlations with actual website users. However, these results omit any users which used built-in website search functions or did not leave the homepage, ignoring information which could be found through state changes but without changing the URL.

3 Main Idea

We will be building a plugin for CrawlJax. The tool will take a start URL (the homepage) and a target URL (the page in question). It will return an ”accessability” rating representing the difficulty navigating from the homepage to the target page.

By default, CrawlJax explored the entire web domain, searching randomly until every page has been explored. We will be implementing more sophisticated techniques that better mimic how humans navigate websites. By implementing our tool as a plugin for CrawlJax, we will be able to construct detailed state-flow graphs of how a user will navigate a website as well as relevant statistics. In addition, thanks to the dynamic analysis functions of CrawlJax, we should be able to take advantage of and properly assess usability of AJAX applications as they are utilized by humans.

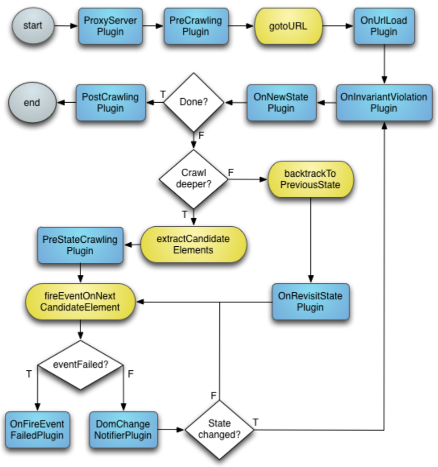

Figure 1: High-level CrawlJax behavior. Entry-points for potential plugins are highlighted in blue.

CrawlJax has robust support for plugins. We will be creating a PreStateCrawlingPlugin. The plugin will override Crawljax’s CandidateElement behavior by implementing our own techniques. We will also be modifying a OnNewStatePlugin to check whether we have reached the target URL, OnRevistStatePlugin to determine a new strategy if a dead end has been reach, and a PostCrawlingPlugin to output relevant statistics.

4 Milestones

There are three Milestones we hope to reach while working on the project.

- Milestone 1 - At it's most basic level, we hope to have a tool that accepts a start and target URL and identifies paths from the start page to the target page. This will involve creating the OnNewStatePlugin, but will not require creating a PreStateCrawlingPlugin.
- Milestone 2 - We hope to implement non-domain specific heuristics to better emulate human behavior. For example, humans are more likely to click on large yellow links. This will involve creating a PreStateCrawlingPlugin, in addition to using the plugin created for Milestone 1.
- Milestone 3 - We hope to implement domain specific heuristics. For example, if a user was searching for the Swarthmore Computer Science page, he would be likely to click on links labeled ”computer” or ”science”.

References

[1] Ali Mesbah, Arie van Deursen, and Stefan Lenselink. Crawling ajax-based web applications through dynamic analysis of user interface state changes. *ACM Trans. Web*, 6(1):3:1–3:30, March 2012.

[2] F Ferrucci, F Sarro, D Ronca, and S Abrahao. A crawljax based approach to exploit traditional accessibility evaluation tools for ajax applications. In *Information Technology and Innovation Trends in Organizations*, pages 255–262. Springer, 2011.

[3] Natalia Negara, Nikolaos Tsantalis, and Eleni Stroulia. Feature detection in ajax-enabled web applications. In *Software Maintenance and Reengineering (CSMR), 2013 17th European Conference on*, pages 154–163. IEEE, 2013.

[4] Ed H Chi, Adam Rosien, Gesara Supattanasiri, Amanda Williams, Christiaan Royer, Celia Chow, Erica Robles, Brinda Dalal, Julie Chen, and Steve Cousins. The bloodhound project: automating discovery of web usability issues using the infoscentπ simulator. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 505–512. ACM, 2003.

[5] Jeffrey Rubin and Dana Chisnell. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests.* Wiley Publishing, Inc., Indiana, USA, 2008.