Final Project - Literature Review

Nathan McIntosh

Due November 17, 2020

1 Introduction

The sources in my bibliography are primarily to educate me on the state of dependencies on Linux and MacOS, and to view successful tools for visualizing dependencies. The first source on "dependency hell" helps me learn patterns I can point out to the viewer. The second on Homebrew on Linux helps me learn about Homebrew's status in the Linux ecosystem. The third on Cargo's dependency tree visualization tool helps me get an idea of what a good, basic tree visualization tool can do. Finally, ndepend's documentation on dependency matrices provides something on the opposite end of the spectrum from cargo-tree.

2 Reference Descriptions

2.1 Dependency Hell [4]

Dependency hell is something that I can address to an extent with my digraph of dependencies. By understanding all of the different forms dependency hell can take, I can more easily point out to the user to the patterns in the digraph that form different types of dependency hell. After reading a basic description of these different types, the user is empowered to make their own observations, gaining a great deal more information from the graph than before they had an understanding of dependency hell (assuming they had no previous knowledge of dependency hell).

2.2 Homebrew on Linux [1]

Something I did not understand well when I began this project was how exactly Homebrew on Linux varied from the default package managers that come with many distributions of Linux. Not understanding Homebrew's standing in this world of Linux package managers means that I would not be able to talk about it properly in its context. This relatively short Quora post explains how most Linux package managers run as root and support system level upgrades, whereas Homebrew is designed to maintain user-owned (as opposed to system owned) packages, and is not run as root. This distinction means that Homebrew has a slightly different purpose, and operates at a slightly different level of software management. However, that is not to say that one might not be used for the purposes of the other. And indeed, it is very likely that many Linux users will get most, if not all, of their packages from their default package managers. This means that Homebrew will likely not have the same ecosystem-wide prevalence that it has on MacOS.

2.3 Dependency Visualization with Rust's Cargo [3]

A language that I am currently learning is Rust, and one of the things that has most impressed me is its robust and mature package management system, Cargo. Cargo is the primary package management and build system for Rust programs. In particular, it has a `cargo tree` command that displays the dependencies of your current rust project. The reference page for `cargo tree` explains a number of ways that dependencies can be displayed, formatted, and investigated. The various input arguments help me think about how I want to structure the display of dependencies in my project, as well as handy options that I might want to give the user, e.g. "show me all the packages that depend on package X".

2.4 Dependency Structure Matrix [2]

ndepend is a tool for the .NET ecosystem offering a number of utilities such as metrics, quality reports, DevOps and CI, and dependency information. As part of the dependency information, ndepend can create a dependency graph and a dependency matrix. Displaying the dependencies as a matrix is a relatively unique method, and has its own benefits and downsides. I may be able to reuse some of these methods in my dependencies visualization.

References

[1] Adrian Ho. *How is Homebrew different from apt-get or Yum?* URL: https://www.quora.com/How-is-Homebrew-different-from-apt-get-or-Yum. (accessed: 2020.11.15).

[2] ndepend. *Dependency Structure Matrix*. URL: https://www.ndepend.com/docs/dependency-structure-matrix-dsm. (accessed: 2020.11.15).

[3] Rust Cargo Team. *cargo-tree*. URL: https://doc.rust-lang.org/cargo/commands/cargo-tree.html. (accessed: 2020.11.15).

[4] Wikipedia. *Dependency Hell*. URL: https://en.wikipedia.org/wiki/Dependency_hell. (accessed: 2020.11.15).