Information Foraging Theory

PROBLEM: WASTED TIME

Software developers often spend a significant portion of their time seeking information within their development environments. Tasks like re-factoring, debugging and writing correctly organized code can involve the developer spending valuable time scanning through their environments looking for examples, regions or specific functions. The time spent navigating through code or seeking information in an integrated development environment (IDE) is not an efficient use of developer time and could be better spent elsewhere.

INFORMATION FORAGING THEORY

Information Foraging Theory (IFT) is the theory and math behind the choices people make to maximize the value of the information they find versus the cost of getting that information. Our aim is to develop a tool that will act as a proof of concept to this idea and increase developer efficiency. Implementing multiple IFT design patterns, we will create a developer tool that helps visualize code structure.

INFORMATION TOPOLOGY IN IFT

In IFT, the information environment is described in terms of a topology. This topology is made up of information patches connected together by traversable links. In our project, information patches will be files and significant code blocks and links will be hierarchical and external relationships.

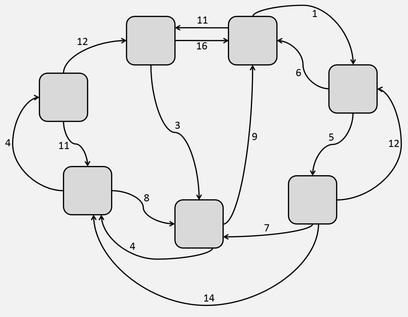

Fig 1. A generic information topology.

IFT GOES POSTAL

Dr. Christopher Scaffidi is a researcher at Oregon State University looking for ways to apply IFT design patterns in software engineering applications. Our team approached Dr. Scaffidi with a proposal to implement IFT as an add on to an existing IDE.

PROJECT POSTAL

A Visualization of Your Code Structure

POSTAL

Postal is a tool to visualize the structure of a codebase using Information Foraging Theory design patterns.

Visualization features:

- Separate, interactive window
- Files and code blocks represented as nodes (ex. HTML div tags or C++ classes and functions)
- Parent-child relationships represented as edges (lines) between nodes
- Links and references between files shown as special edges
- Configuration options including topology shape, link toggling and visualization “physics”

Postal is an open source, cross-platform extension, developed and tested on:

- Windows 7 and 10
- Linux (Ubuntu)
- MacOS

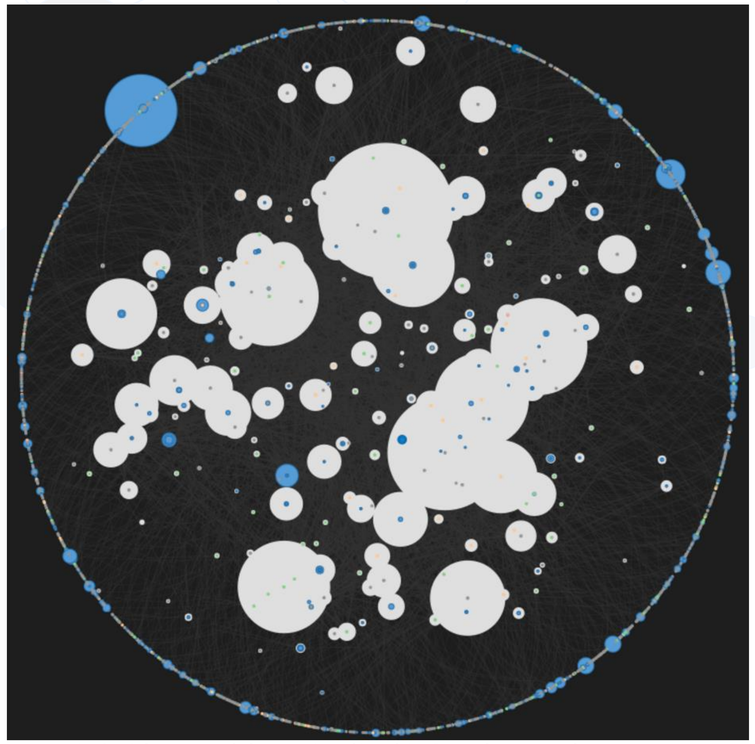

Fig 2. A large codebase visualized in Postal as a web structure.

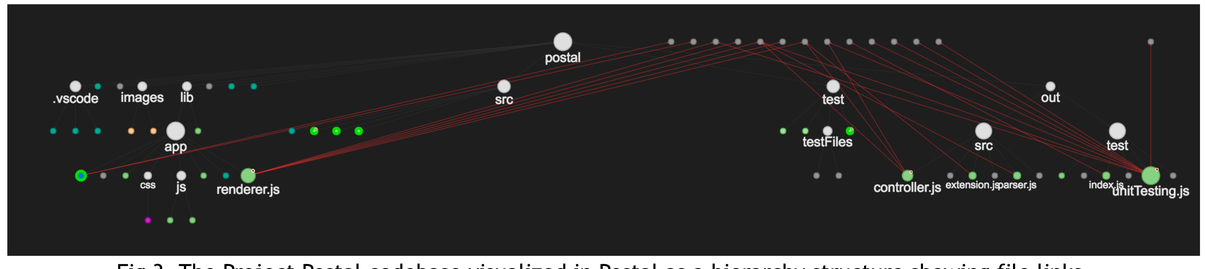

Fig 3. The Project Postal codebase visualized in Postal as a hierarchy structure showing file links.

APPLICATIONS

Postal is a configurable tool with a variety of applications. Through manipulation of Postal's grammars Postal can perform:

Website Mapping
Postal is capable of mapping websites and displaying links between directory files and external sources.

Visualization of Specific Function Calls
Useful for seeing what code is used where and how frequently it is used.

Project Standards Enforcement
Using the notification system, users can write grammars to find bad practices in code that might not be wanted.

VISUAL STUDIO CODE

- Created by Microsoft
- Released April 2015
- Extension marketplace launched 2016
- Open source, cross platform, free

Creating a VS Code extension allowed our team to focus on the tool itself and not have to worry about how the user would invoke our tool. It also gave us an excellent platform to which to market our extension. Having this tool up on a marketplace meant we didn’t have to distribute it ourselves and we could focus on building the tool and marketing it towards our target audiences.

Senior Capstone Project
CS 461/462/463

About Team Postal

Team Members (from left to right)

- Eric Winkler, Computer Science
  winkleer@oregonstate.edu
- Sam Lichlyter, Computer Science
  lichlyts@oregonstate.edu
- Cramer Smith, Computer Science
  smithcr@oregonstate.edu
- Zach Schneider, Computer Science
  schneidz@oregonstate.edu

Sponsoring Client

- Christopher Scaffidi, Associate Professor, Computer Science
  Oregon State University
  scaffidc@eecs.oregonstate.edu

Languages and Technologies

- JavaScript
- TypeScript
- NodeJS
- Electron
- Visual Studio Code

User Testing

- User testing proposal submitted to IRB
- Results will determine effectiveness of Postal compared to base VS Code text editor

Oregon State UNIVERSITY