

A Tool to Automatically Organize the Structure of a Codebase Using Information Foraging Theory Design Patterns

Progress Report

Team Postal — Group #38

Cramer Smith, Sam Lichlyter, Eric Winkler, Zach Schneider

CONTENTS

I	Project Purpose and Goals	3
II	Current Status	3
II-A	Parsers	3
II-B	User Interface	3
II-C	Data Layer	3
III	Problems and Solutions	4
III-A	Pre-Week 3	4
III-B	Week 3	5
III-C	Week 4	5
III-D	Week 5	5
III-E	Week 6	5
III-F	Week 7	6
III-G	Week 8	6
III-H	Week 9	6
III-I	Week 10	6
IV	Prototype Code Samples	6
IV-A	FileMap UI	6
IV-B	HTML Parser	6
V	Fall Term Retrospective	7

I. PROJECT PURPOSE AND GOALS

Developer tools are often complex pieces of software. Gathering and manipulating useful information for a programmer can often be a slow and costly process. By implementing Information Foraging Theory design patterns in the creation of these tools, the information collected may be more useful or obtained faster. 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. The aim of this project is to develop a tool that will act as a proof of concept for IFT and increase developer efficiency.

The Postal extension is being designed to allow developers to more quickly search through and better visualize their projects. This extension will also help developers create clearer and cleaner code structure by offering reminders and suggestions about the best coding practices in the programming language they are currently using. Any major errors or incompatibilities within the project or its files will be reported to the developer as well.

II. CURRENT STATUS

As of the end of Fall term, the Postal team has completed all documents and preliminary requirements needed to move forward with the creation of the Postal extension. Initial prototypes for both UI elements and file parsing components of the extension exist and are beginning to be further developed. The team aims to have a fully working prototype by the end of Winter break that is able to parse and display any web development project. Current and prior progress indicates that the Postal team is on track to meet this self and client imposed deadline. Below are current descriptions and functionalities provided by the major components within the Postal extension.

A. Parsers

The current implementation of the extension has one parser which parses HTML. This parser is implemented in Perl using the Simple TokenParser which looks for specific tags and a specific attribute from that tag. For example, this HTML parser specifically looks for the anchor (`a`) tag and the hyperlink reference (`href`) attribute and returns those values which are the links to either outside websites or internal pages.

B. User Interface

The user interface is currently running off of dummy, static data. It opens up in a new window using a program called Electron as opposed to a web browser to cut down on the overhead associated with all the complexities of the web browser. It shows each web page as a node and each link from one page to another as an edge in a graph. See Figure 1.

C. Data Layer

The data layer is comprised of a data structure and data access methods. The data structure is defined below in the ERD (Figure 2).

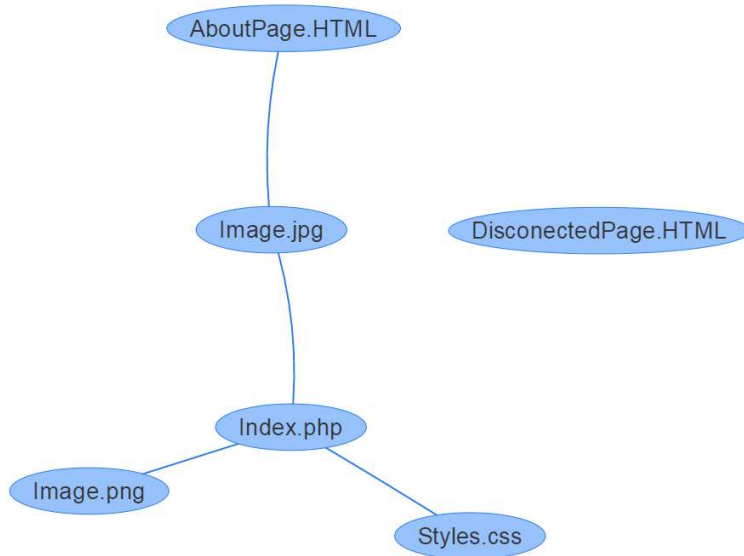


Figure 1. User Interface Example

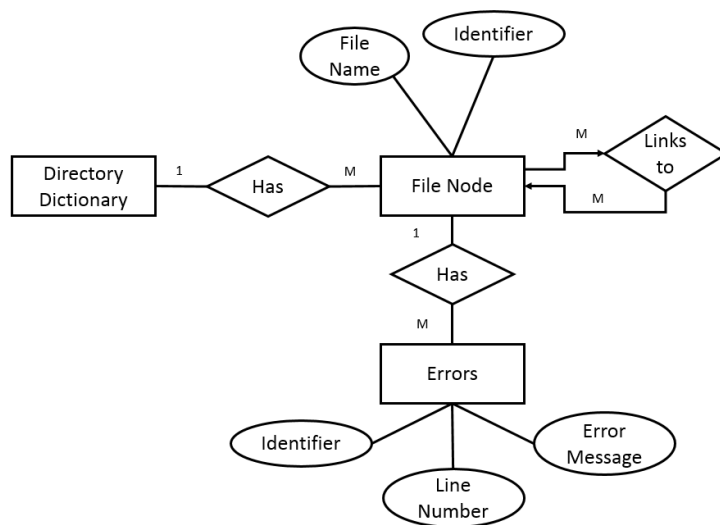


Figure 2. Data Structure ERD

III. PROBLEMS AND SOLUTIONS

A. Pre-Week 3

The beginning of our project was back in August when Sam told the rest of the team about an opportunity that he had to work with Prof. Chirs Scaffidi on a senior design project. The members of the team all jumped on the opportunity to work together for a professor all of us had had previously and thoroughly enjoyed. Prof. Scaffidi tasked the team with coming up with project proposals. The only requirements for the project is that it had to be a

tool that used information foraging theory to do anything. The first problem was that none of the team knew what IFT was so the team took a week to brainstorm separately and research IFT and come up with project ideas. After that week the team met and decided on two ideas to give to the client. Scaffidi chose a code base organization tool that would be an extension for Visual Studio Code.

B. Week 3

After the first couple weeks of class while other teams were just meeting the Postal team was able to start meeting and thinking about functionality as well as begin to work on the problem statement. The problem statement was the first document of the project and it was a challenge to create an official document and learn how to use the IEEE standard. This problem was not really solved. The team ended up creating what they thought was a good document, and there was feedback on the document that came later in the quarter helped tremendously for the next draft of the document.

C. Week 4

During the fourth week, our client had challenged the team with accelerating the development process past the class. In response to this request the team started developing basic "hello world" extensions with very limited to no functionality. These programs served as a learning experience for the team as we develop more complex extensions these examples gave us a framework of understanding of what the code of an extension looks like. These small programs were good but not up to par with our clients expectations.

D. Week 5

In week five the team revised the problem statement. This task was fairly easy as the team had received feedback on the first draft and therefore had a knowledge of what was expected of them. This week we also started work on the Requirements Document. The requirements document was much more daunting than the problem statement as it was more technical, and made the team think more the project as a piece of software and not just a problem to be solved. The team wrote a first draft of the requirements document and made a plan to finish it the next week.

E. Week 6

With the completion of the requirements document the team had some solid ideas as to what the project was going to look like. This was a big step forward for the project as the team had only a fuzzy idea as to what the project really looked like at this moment. Now the team knew that it was going to be a web development tool instead of a general code organizer. This was great for the development of the project, but this narrowing of scope was not communicated to the client. The team thought that they had communicated this in a meeting but apparently either our client did not understand or the idea was not stated clearly. This caused problems later in the process.

F. Week 7

This week we thought was a big step for the team. By this time there was a working prototype of the user interface, and a simple parser for HTML. The only thing was that these programs were not actually a Visual Studio Code extension but separately compiled programs. The team needed to figure out how to combine the program UI with Visual Studio Code.

G. Week 8

The eighth week the team finished the tech document. The tech document made the team look at alternatives for parts that each member had decided to focus on. This was eye opening as we researched alternatives to pieces of the project that the team members had never thought of before.

H. Week 9

This week was Thanksgiving week. With the extra time Sam and Eric were able to further the prototype and make some small improvements. These small improvements still had not added the UI to the extension but the team thought that it would be good enough to show to the client.

I. Week 10

Week ten was a big week as the design document was due that week. This document proved to be the most confusing as the team had a particularly difficult time figuring out the IEEE standard of the software design document. The team was able to struggle through the design document and create what the team thought was good document. The team also was able to show to the client what we had done with the prototype so far. The client was happy with the prototype, but encouraged the team to reconsider the reduction in scope that had occurred previously. This will be factored into the work going forward.

IV. PROTOTYPE CODE SAMPLES

A. FileMap UI

B. HTML Parser

```
1 | my $anchor_parser = HTML::TokeParser::Simple->new(handle => *DATA);
2 | while (my $anchor = $anchor_parser->get_tag('a')) {
3 |     next unless defined(my $href = $anchor->get_attr('href'));
4 |     say $href;
5 | }
```

V. FALL TERM RETROSPECTIVE

Topic	Positives	Deltas	Actions
Communication with client	<ul style="list-style-type: none"> • Flexible in terms of project requirements. • Provided the team with a mock up application architecture. • Easy to work with. 	<ul style="list-style-type: none"> • Frequency and clarity of communication need to be improved. 	<ul style="list-style-type: none"> • Solidify team member responsibilities (Sam is in charge of emailing the client). • The team as a whole will need to make sure Sam is aware of things that need to be communicated with the client.
Team Effectiveness	<ul style="list-style-type: none"> • Members have flexible schedules and are easy to sit down with. • Members all do a good job of quickly replying to communications. • Members have a positive attitude. 	<ul style="list-style-type: none"> • As a team, assignments do not get much planning and are usually done last minute. • There is no clear team leader. 	<ul style="list-style-type: none"> • Assign a pseudo team leader with the responsibility of tracking/ scheduling tasks and work assignments.
Implementation	<ul style="list-style-type: none"> • The team has become much more capable at developing with the VSC IDE. • The Team has already built a couple of working prototypes for the client. 	<ul style="list-style-type: none"> • The Gantt chart is currently out of date and needs updating. • Some technical implementation obstacles have yet to be addressed (npm auto installing). 	<ul style="list-style-type: none"> • Update Gantt chart to better reflect the current progress and set new goal deadlines for winter term.
Design	<ul style="list-style-type: none"> • The current design meets requirements and team members are aware of their design related responsibilities. 	<ul style="list-style-type: none"> • IFT Design Pattern tie-ins are not 100% clear and as a major requirement of the project, they need to be. • Parse needs to be redesigned to better match our scope. See Week 10 notes. 	<ul style="list-style-type: none"> • The team will need to meet to re-design the parser component of the extension and review the selected IFT design patterns.
Documentation	<ul style="list-style-type: none"> • All assigned documentation is currently in a workable state. 	<ul style="list-style-type: none"> • Team members feel that the documentation is typically rushed and not as descriptive as it should be. • Team members have a difficult time interpreting the IEEE standards and this may be a partial cause to the above. 	<ul style="list-style-type: none"> • A team member will take a leadership position to schedule work tasks which will ensure that the team has sufficient time to complete the documentation assignments to a higher standard.