Initial Plan: A Syntactic Pattern Recognition Approach to Wave Analysis

Author: James D Bateman (C1884464)
Supervisor: Dr Paul Rosin
Module Number: CM3203
Module Title: One Semester Individual Project

Project Description

This project aims to apply the syntactic method of parsing to the analysis of water waves. More specifically, this project uses the syntactic method to analyse waves in order to determine their adherence to bounding factors like amplitude and period. The project will produce software that is capable of building a grammar based on vessel data, and then using that grammar to parse waves as they would hit a vessel.

The general method for analysing water waves uses the statistical method of wave analysis. Statistical wave analysis is useful and commonplace, but produces results which are not easily understood in context. The syntactic method of wave analysis is little explored but has produced promising results in the field of ECG trace analysis. The output produced by a syntactic analysis is semantic in nature, that is to say that it conveys contextual meaning. In the context of wave-on-vessel analysis, contextual semantics could provide useful and readable data that would assist vessel captains. This project explores the syntactic method over statistical recognition due to the syntactic method's ability to carry semantic and structural information which the author hopes will be useful to vessel captains as they assess the safety of their vessels.

I aim to provide a means of better understanding water waves in relation to vessel safety by producing a software package that is capable of building a function, specific to a vessel, and using that function to parse a set of waves as they would hit that vessel. It is hoped that by parsing waves as though they were a language, we will be able to produce data for vessels that improves vessel safety and allows vessels to gain a greater understanding of how close they are to being in dangerous conditions.

Project Aims and Objectives

Aims
This project aims to provide a means of better understanding of water waves in relation to vessel safety by producing a software package that is capable of building a function, specific to a vessel, and using that function to parse a set of waves as they would hit that vessel. I aim to successfully implement this software package over the next 12 weeks by building on a prototype that I will construct in the first 2 weeks. A successful software package should be capable of taking as input a JSON representation of a vessel and producing a function which is capable of parsing waves as they would hit that vessel, producing as final output a semantically rich analysis of a waves proximity to bounding safety factors. The software should produce well-formed outputs from syntactically correct inputs, and generate appropriate and useful error messages when passed syntactically incorrect inputs. The software should also be readable and usable outside of the command line, such that vessel captains could make use of the system.

Objectives
In order to achieve my aims, I will produce a piece of high quality software that performs the function of a parser generator. Specifically, I will develop several component modules which will come together to form a software package that addresses the requirements of my aims. In order to do this I will need to research background ideas and investigate the methods employed by those who have attempted similar challenges to determine the best way to develop my components. I plan to iterate on an initial solution by developing a prototype to assess feasibility, then continue to add modular functionality to the program until it meets my requirements. I aim to operate in an agile manner; first producing a working MVP-style prototype, and then producing individual component modules that are able to plug onto the prototype. This way, I hope to maintain a working product throughout development instead of working towards a monolith that would only work when entirely complete.

Project Components
Deliverable I: Wave Parser Generator

- Wave Parser Generator Prototype: The first two weeks or so will involve the implementation of a feasibility test for the wave parser. This prototype will consist of a simple grammar and parser, designed to take as input a synthetic sentence of symbol representing some water wave data.
    - Define a basic Grammar and Parser
    - Synthetic Wave Tokens
    - One or Two vessel variables
- Error Module: Robust error checking and reporting are a major part of a parser. After writing this module, the program is now able to distinguish between program errors, such as a malformed input or an internal error within the program, and different types of parsing errors. In this implementation, parsing errors represent a dangerous environment for a vessel and must be human-readable on the command line and machine-readable as input
    - Program Errors
    - Parsing Errors
- GUI: Write a GUI that allows people to use the program on a vessel or in a lab without the command line. A GUI is a more friendly means of using this program. This part of the project is dedicated to polishing a presentable GUI that displays the input and output of the program for users to view. This GUI will be able to hook into events that the program fires in order to present a live feed to the user.
- Tokenizer: Write a wave tokenizer, allowing the wave parser to take as input real water wave data; tokenize it, and use it as input for a parser.
    - Delineate and define functions for chunking waves
    - Write a function to take wave data and tokenize it.

Deliverable II: Report

- Report: Allow time for collating material learned and writing up the report.

Expected Results
A program capable of parsing wave data based on an initial input of vessel data, producing output that determines if the waves are safe for the vessel and how close to being safe or unsafe the waves are in a semantically meaningful way. I.e: “Wave period needs to be 0.2s shorter to be unsafe.”

Work Plan

Weekly Plan

Project Weekly Plan																			
Progress	First Iteration, Proof of Concept				Parser Works, Output Given			Improving and Expanding			A Finished Project			Evaluating the Result			Finalising and Writing Up		
Weeks	Week 2	Week 3			Week 4			Week 5		Week 6	Week 7		Week 8	Week 9		Week 10	Week 11		Week 12
Module	**Prototype**			**Meeting**	**Write Error Module**		**Write GUI**	**Wave Tokeniser**			**Parser Generator and Parser**			**Meeting**	**Evaluation**			**Finalising**	
Breakdown	Define Grammar and Parser	Synthetic Wave Tokens	Few Vessel Variables		Program Errors	Parsing Errors	HTML Output from NodeJS	Packetise Waves		Tokenize Waves	Define JSON Vessel representation, add more functionality to PG				Evaluate the program.			Collate material and finish report.	

Week 2 & 3

In weeks 2 and 3 I will develop a prototype Wave Parser Generator, consisting of a simple grammar and parser. This prototype will not take arbitrary vessel data as input, but instead will focus on creating a parser generator which looks only at a few vessel variables such as length or breadth. The parser generator will produce a parser that can then take as input a synthetic sentence of symbol representing some water wave data. I intend to meet this milestone by first researching and experimenting with known parsing techniques to see which is best for a left-recursive structure like a wave/time input, then the prototype to implement what my research has led me to. At the end of week 3 I will have a meeting with my supervisor to make sure that the prototype works and that a more featureful iteration of the prototype is possible and feasible, which will inform me on my actions as I move forward. By the end of week 3, I should have a working prototype that will function as a framework around which I will continue to develop my project.

Week 4

In week 4 I will focus on developing an error-checking module that enables the Wave Parser Generator to give sensible and readable output that could be used in context. A parser must distinguish not only between program and parsing errors, but also between all the types of parsing errors that can occur. In this sense, the error module of a parser is right at the core of the program. In some ways a parser could be said to be a complex error-checking module. I Intend to meet this milestone by programming and testing code that will serve to function as the error-checking functionality for the software. Also in week4 I will develop a basic HTML-based GUI that will allow the program to be interacted with visually and produce output that is interpretable visually. I intend to meet this milestone by using ExpressJS, a well-known framework, to allow the internal data in the program to be output to a browser via HTML. By the end of week 4, the software under development will have a GUI for ease of use and a robust parsing output which will allow the user to determine how their wave data has been parsed.

Week 5 & 6

Through weeks 5 and 6 I will turn my attention to developing a wave tokenizer that is able to turn water wave data into a parsable form. This major step will mark a transition in my project as the software will now be able to take as input real wave data, rather than synthetically generated tokens meant to represent waves. I intend to meet this milestone by conducting research into the methods that other people have employed to tokenize wave data and implementing a method that I will narrow down based on my research.

Week 7 & 8

Weeks 7 and 8 will see me further develop the Parser Generator. During this time I will develop a JSON structure which all vessel data will subsequently use to format their input. Using this structure I will add functionality to take advantage of more pieces of vessel information, meaning a more refined parser is generated against more varied input. I intend to meet this milestone by incrementally adding vessel information to the input of the Parser Generator, with each addition meaning that the Parser Generator is producing more refined parsers. At the end of week 8, I will have a meeting with my supervisor to show him the progress of the program as it stands at this point.

Week 9 & 10

I will spend weeks 9 and 10 evaluating my softwares output against known datasets in order to determine how successful and useful it is.I intend to meet this milestone by comparing the output of my program against reference vessel/wave data provided by British Marine Organisations. This will allow me to determine if the program can indeed tell that a wave is dangerous for a vessel.

Week 11 & 12

During the last 2 weeks of the project, 11 and 12, I will be finalising my software and writing up my report. This might entail fixing small bugs or implementing small additions along with the task of actually writing my notes and thoughts into a report. I intend to meet this milestone by collating the knowledge that I will be discovering, bringing together notes and source code, and making conclusions about the project which I will have undertaken.

Deliverables

At the completion of this project I intend to deliver a report that details my progress and findings. Included in my report will be the software package which I have developed, which will be capable of creating a parser from vessel data and then parsing wave data against that parser.