YAPL Design Document

Objectives:

The objective of this project is to create a programming language to allow programs to quickly recover from errors during their execution. It will do this by being deterministic, meaning its next action is predictable. This would allow a separate server to pick up execution at a point when the program was executing correctly. It will also be compatible with anything that can compile and run C code, because it will convert the program written in this language into C code. It will also be simple to learn and to use, and make it easy to crate a deterministic program, without having to think about too much.

Users and Use Cases:

The only person who will use this programming language is the programmer himself. They are the one who will be tasked to create the program that is being written. Those using the program will not know that they are even using a program written in this language.

We'll call this programmer Bob. Bob wants to create a program to fly a drone, but that drone is very valuable so he does not want anything to happen to it. Bob decides that instead of using a standard language he wants to use a language that was made specifically to work in this situation. First Bob will have to decide what he wants the drone program to accomplish. Next Bob will have to determine a set of states, that when put together will accomplish this task, as well as maybe some fail states, that can recover from errors. Now Bob will be able to begin writing the code. Once his program is written he can compile it using the YAPL compiler. This will then convert the YAPL program into a C program, and compile that into an object file that can be run.

Bob can also use his new program with another piece of software that can help to fix the program if there is another error. All he has to do is run this program on a server with the error recovery program running and have another server also running the error recovery program.

YAPL PROGRAM

LEX FILE

YAPL COMPILER

YACC FILE

C PROGRAM

OBJECT FILE

The Compiler is a combination of a lex file and a yacc file working together. The lex file returns values relating to the key words it finds in the YAPL file. And the Yacc file determines patterns in the YAPL file to determine if the syntax of the file is correct. Whenever the YACC file determines what it is parsing in the YAPL file, it is able to act. It converts each part of what it finds into C code. This C code will perform the same actions that the YAPL file was intended to perform.