Project Report

**Introduction:**

In this report we address the problem of that there’s no support for aspect oriented programming in Javascript.

The approach to solve the problem is a library that Javascript users can use in order to get aspect oriented features in Javascript. I’ve used Babel the Javascript compiler or parser. I’ve used it to create a simple library that runs and takes as input two files. One is the application file, and one is the aspects file. It uses a simple API, the user is expected to use a weave function to indicate which functions he wants to weave into which functions. The library looks for the weave function calls, and weaves the aspect into the function. The library also supports objects. The library can weave an aspect into all functions of an object.

The structure of the code is simple. There’s a main.js file which takes as input both application code and aspects code. It outputs new code of the aspects woven into the functions.

We compare our solution to python’s aspectlib and java’s aspectj. Aspectj can be a bit different since java is a strongly typed language and an object oriented language. I haven’t used decorators in any of the implementations. I’ve used an external file that works on the main code files without modifying them.

Aspectj uses its own language while Aspectlib relies on python for the implementation. My library uses javascript with some tricks or some API. Strictly thinking it can be thought of as a different language that’s in the style of Javascript.

In aspectj the program is run as an aspectj application. In aspectlib it’s a library that can be imported and used in python. In my library the entry point is my library itself. So the approach can be thought to be similar to Aspectj’s solution.

One of the challenges is that the documentation for Babel is a bit scarce and is not very descriptive. I can’t find a lot of examples. I contributed to questions on stackoverflow. I solved it sometimes by trial and error. I solved it sometimes by inference. Just look at some similar functionality and try to infer from it what I can do to achieve my goal. Another challenge is that I’m only or mainly relying on babel to make my library. I’ve used only a few extra libraries to do some basic operations. I’ve used fs to read from a file. I’ve used clone to make deep clones of javascript objects.

Usage:

npm install

node main.js file1.js file2.js

where file1.js contains unmodified javascript code of some application

and file2.js contains aspects to be woven into the code.

to run the test cases run:

node main.js testcase1.js testcase1client.js

node main.js testcase2.js testcase1client.js

also there's the play area folder. It contains an example for aspectlib in python. Run:

source venv1/bin/activate

python3 testcase1client.py

python3 testcase2client.py