Project Report

The problem was that there’s no support for aspect oriented programming in Javascript.

The approach to solve the problem was a library that Javascript users can use 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 uses is expected to use the same function name for his aspects. The library looks for the aspects and functions that have the same name 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.

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.

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