I downloaded Java-WebSocket from <https://github.com/TooTallNate/Java-WebSocket>. It is a simple WebSocket client and server implementation written in Java. It allows for easy implementation of sockets for java applications, if someone doesn’t want to use the default implementation. It’s mostly a toy example written so the authors could have a better understanding of sockets. There are 2,245 Lines of Code.

The Design Pattern Finder tool looks at links between objects in the code and analyzes them to see how they match up against the patterns it knows of. This process is adequate for analyzing a design after the fact. It’s far from perfect; it won’t catch everything and will flag patterns that aren’t there. However it seems decent in finding patterns. If I was writing a similar tool I think I would likely try the same approach. If I was feeling really ambitious I may try writing an AI agent using a Bayesian network and teaching it various examples and counter examples of various design patterns to see if I could do better, but that would be a lot of work.

The realization I studied was ServerStressTest using a State Pattern (See attached derived diagram of that realization). The realization is a true realization, it matches the generic pattern exactly. The realization makes use of the pattern to handle the various states that a connection can take (connected or not connected). It was not difficult to compare as it matched so exactly. If the match had not been this easy, I would have compared the number of links and how the various components were coupled together then made a ‘best guess’.