Ahram Kim

Mark Bastian

CS 498

February 23, 2018

Seminar 6 (02.23.18)

http://github.com/markbastian/reloadable

Mark Bastian was giving a talk this Friday. He is a senior software developer for Clearwater. The title is “Interactive Development with Reloadable Code”. The word of reloadable code is that code changes are immediately propagated into a client upon save without also resetting the client state. This result is a dynamic and immediate feedback loop which allow for rapid prototyping and development.

The coding cycle is code, save, compile, execute, set state, and close. The motivating example of developing a temperature is to start with basic frame, run it, add UI elements, run it, add model about getvalue(), tofarenheit(), and tocelsius(), implement model, and additional steps about listener, propertychangesupport, and restoring state and debugging.

Reloadable code is application in a long-running process, change code, trigger refresh, changes are propagated, no manual client refresh, and los of state. Facilitating reloadability could be interactive environment, and separation of concerns for value, behavior, and state. The solution for Figwheel is that Figwheel builds you clojurescript code and hot loads it. Also, the solution for Quil is that sketches can be run as java or javascript apps.

The reloadable code was unfamiliar for me so that seminar was interesting. That is the nice chance to take the seminar from the people of real company and real software engineering environment. However, his seminar was kinds of shorts as I thought. That was the short time that I can understand and figure out what the reloadable code is, and how it works. After the talking, he brought the code for the reloadable code and showed how to execute that code. That made me understand well but still need to study for code. It looked that reloadable code is interactive, productive, and fun!