angular.html: https://github.com/marceloarchiza/cs462lab5/blob/master/angular.html

temperature.js: https://github.com/marceloarchiza/cs462lab5/blob/master/temperature.js

wovyn\_base.krl: https://github.com/marceloarchiza/cs462lab5/blob/master/wovyn\_base.krl

sensor\_profile.krl: https://github.com/marceloarchiza/cs462lab5/blob/master/sensor\_profile.krl

1. What design decisions did you make in your rulesets that made this assignment easier or harder? Why?

Last time I was accessing the global variables with events instead of querying them. I learned how to query them instead and it made my life easier. Also having a query that retrieves the entire profile made it easier than querying each global variable at a time. So I could just query for the entire profile to display on my website.

1. Explain how the sensor\_profile ruleset isolates state and processes regarding the sensor profile from other rulesets.

It isolates because none of the rules have the same path from the other rulesets. It has only one path that lets the user update the profile and it stores the profile. Other rulesets can use the profile or not, but the profile is isolated and if anything breaks elsewhere, the profile does not need them, and it will still work. So, it is also independent.

1. How do other rulesets use the sensor\_profile to get data?

They import the ruleset and since the profile provides all the global variables/functions, the rule set that imports the profile can access any of those variables by simply calling the methods.

1. Could they use it to store new values? How?

Other rulesets can use profile to store new values by raising the sensor/profile\_update event. This event can be accessed because sensor\_profile shares the profile\_update rule.s

PS.: On the video I mentioned that there was a bug on my code. I was not displaying the correct current temperature. I fixed the bug and here is a screenshot of it working properly:

