Recitation #6

18-649 Embedded System Engineering
Friday 9-Oct-2015
Announcements and Administrative Stuff

◆ Project 5 due yesterday

◆ Project 6 is posted

◆ Project 6 is due Friday Oct. 16th by 10:00 PM
Minimum Requirements Document

- Project is not turned in until a COMPLETED minimum requirements chart for your group is filled out
  - This includes the hours spent since last project

- You will accrue late penalties until this is turned in
Reminder: Java Files

- **All your code belongs in the elevatorcontrol package**
  - Including your payload translators (if you wrote them)
  - This is where we place the files from your *portfolio/implementation* folder

- **Java files need to compile on the ECE machines**
  - No dependencies on weird libraries.
Build Teams (Assign this role to a team member)

- Build Teams in software development in the industry ensure –
  - All the modules are the latest
  - The code does Clean compile
  - The Watchdog timer is working
  - Final build passes tests one more time

- You have a build process too
  - Must be assigned to one person explicitly (should be clear who has to do it)
  - Look at the sitemap for scripts to help with this
  - Ensure that the project compiles (all Code and Test)
  - Check the Project against the grading rubric (including re-running the tests)
  - Run the code on the ECE machines

- Compilation is **23% of your grade** for project 6
Project 6 - Overview

- More of the same from project 5

- Implement second half of elevator
  - Dispatcher
  - Lantern Control
  - Car Position Control

- Traceability - State chart to code

- Unit testing

- Integration testing
Implementation

- **Create new java files to implement four controllers**
  - Place these files in ../simulator/elevatorcontrol/
  - Each module must be included in simulator.elevatorcontrol package

- **General requirements listed on the website. Some examples:**
  - You shall use the interface defined in the behavioral requirements
  - You shall NOT add additional communication channels between controllers
    - No accessing global variables, etc.
    - Just communicate using network and physical messages
  - You shall adhere to the message dictionary and interface
    - Don’t be tempted to create new messages or modify the dictionary

- **We’ll eventually run your implementations on our own test files**
  - Probably fail tests if your design uses secondary channels or altered dictionary
Traceability

- All transition arcs must be traced to the code that causes the transition
  - In most cases, comment just above the if statement that tests guard statement

- Code must contain comments that indicates each transition
  - Forward traceability

- Portfolio must include traceability table
  - Each transition and its corresponding code line # must be in the table
  - Backward traceability

- Detailed instructions and hints on project 5 web page
Testing

◆ Project 5 page contains link to detailed instructions for testing
  • You must perform each step listed in the detailed testing instructions

◆ Unit Tests
  • Exercise all the transitions in your state chart
  • Reminder: If your transition has an OR, you must test both branches!
  • You must pass all unit tests for all controllers

◆ Integration Tests
  • Select *TEN* sequence diagrams
    – Must include specific scenarios (4A, 5B, 6A, and 9A)
    – OK to include the two from Project 5 in this set
  • Must pass *EIGHT OUT OF TEN* integration tests

◆ Traceability required for each test
◆ Peer review required for each test (unit and integration tests) and for each module that is implemented (code).
Questions?