

Recitation #7

18-649 Embedded System Engineering

Friday 10-Oct-2014



Note: Course slides shamelessly stolen from lecture
All course notes © Copyright 2006-2011, Philip Koopman, All Rights Reserved

**Carnegie
Mellon**

Announcements and Administrative Stuff

- ◆ **Project 7 – Midterm Acceptance Testing posted**
- ◆ **Project 7 is due Thursday Oct. 18th by 10:00 pm**
- ◆ **Check-list to use prior to e-mailing staff posted on admin page**

Project 5 Notes:

- ◆ Drive is (Speed s, Direction d)
- ◆ What is DriveSpeed?
 - ◆ (Direction d, double s)
- ◆ Who sets DriveSpeed?
 - ◆ Architecture diagram
- ◆ mDriveSpeed?
 - ◆ How can we stick a double in bits?

If you need some examples of how a part of the simulator works or how the portfolio should look, refer to the

SODA MACHINE EXAMPLE

Project 5 Notes:

- **Tran**

E-mail Check-list (On Admin Page)

◆ Before writing that e-mail

- Check blackboard to see if an answer has been posted
- Re-read the assignment to make sure you are reading it correctly
- Look at the grading checklist to see if it has relevant information
- Look at the Pepsi machine example to see if it provides a reasonable example
- Discuss the problem with your teammates and see if you can agree upon a reasonable way to proceed without violating written assignment requirements

◆ Regarding e-mail on assignments

- *If you simply don't understand, then skip the e-mail and go to office hours*
- If you think there is a defect in the course materials, include the URL of the document you have a question about and a specific explanation of the defect or contradiction
- Start your e-mail with "I've used the e-mail question checklist, and I think the following is an issue:" or the e-mail might not be replied to
- Wait 5 minutes before sending. Seriously. We get lots of "oops, found it" e-mails less than 5 minutes after sending a query

Submissions

◆ Make sure you follow the submission guidelines

- Do not put your portfolio down in an extra sub-directory
- Do not submit a zip file containing your portfolio
- Do not include SVN files
 - Remember to export your portfolio before submitting
- Follow the directions. Don't make extra work for the TAs
 - The above may sound trivial, but it all adds up
 - Extra work for TAs makes it difficult to get grades back to you in a timely fashion
 - Penalties for incorrect submissions will also increase over the semester

◆ Submission guidelines are especially important for code submissions

- We want every .java file in your elevator.control folder
 - Including Control.java and any translators your group created
- We'll be dropping the contents of your elevator control folder into a clean copy of the elevator framework for testing
 - If you stored your files in some creative place, your code will probably break
- List the java files in the appropriate HTML portfolio page
- All .mf files must be included in your test submissions i.e. defines.mf

Project 7 - Overview

- ◆ **Organize Portfolio**
- ◆ **Complete Integration Testing**
- ◆ **Develop and Use Runtime Monitoring**
- ◆ **Pass an acceptance test**

Organize Portfolio

- ◆ **Make sure your portfolio follows the structure and guidelines we have laid out on the portfolio page.**
 - ◆ Check out the Soda Machine Example
- ◆ **If you have been doing this all along, this should be no problem.**
- ◆ **Make sure to fix any problems that have been pointed out in your TA meetings and/or project feedback.**

Complete Integration Testing

◆ You must test all your sequence diagrams.

- If you have more than 20 sequence diagrams, you only need to test up to 20
- Must include original scenarios included in portfolio (see project write up for more details)

◆ Your elevator must pass all the integration tests.

- This is naturally a step toward passing your first acceptance test.

◆ Peer review

- This week you must peer review at least 4 of the newly created integration tests for this project.
- We highly encourage that you do more than the minimum to catch bugs in testing.

Runtime Monitoring

- ◆ **Use the runtime monitoring framework to study performance**
 - Simulator.framework.RuntimeMonitor contains documentation
 - Provides even-triggered access to the physical state of the entire system
 - Messages received by RuntimeMonitor will trigger a callback function receive() giving you the ability to distinguish between messages, log, and report the proper system performance.

- ◆ **Two sample monitors have been provided as a starting point for you**
 - SamplePerformanceMonitor – Door Statechart & weight sensor
 - SampleDispatchMonitor – Fast speed checks
 - Implementation non-intuitive - use these as a starting point!

- ◆ **Read documentation of simulator.framework.RuntimeMonitor**
 - Generate doxygen by typing “make” from the root of the project template

Runtime Monitoring

◆ Implement your own monitor that must:

- Count the number of times elevator becomes overweight
- Count the number of wasted openings
 - An opening is wasted if the doors open when there is no pending call at that floor
 - We are expecting this to happen a lot since you are designing a sabbath elevator
- Count amount of time spent dealing with door reversals

◆ Peer reviews

- Runtime monitor code must be peer reviewed.
- The Runtime monitor won't do you any good if it doesn't work

Acceptance Testing

- ◆ **This week you must pass one basic acceptance test that we provide you.**
 - Instantiates the entire elevator control system with simulated passengers
 - Encouraged to make your own tests. Documentation on how to create acceptance tests is on the course webpage.
 - BONUS: Passing the 2 additional acceptance tests we provide you can net you up to 10 points **IF the rest of your portfolio is consistent and on time**
- ◆ **Keep an accurate log of any acceptance tests that you run.**
- ◆ **Make sure you use the runtime monitor that you implemented while you run your acceptance test.**
- ◆ **Again you shall pass proj7acceptance1.pass this week.**
 - If you do not pass it this week you shall eventually pass the test to receive a grade for the course. If this is the case, contact course staff to arrange a demonstration when you are ready.

Automation

- ◆ **This is a good time to start looking into automating some parts of your project**
- ◆ **We will be providing you with a simple script to get you started**
 - **Feel free to add to this script to match your teams individual project**
- ◆ **Things you may want to automate:**
 - **Exporting your code and making sure it will build for the TAs**
 - **Running all Unit Tests/Integration tests**
 - **Copying *.java files into your portfolio**
 - **Updating statechart to code traceability**
- ◆ **Important Notes:**
 - **Don't automate things that are faster/easier to do by hand!**
 - **Automation scripts can introduce more bugs into your project.**
 - **Just because the script runs doesn't mean it ran correctly and does not guarantee it will run correctly for the TAs.**

In Class Presentations After Break

- Course web site has description of contents

Read it carefully

Follow the format requirements, especially font size

Note that the controller you talk about depends upon your group number

Slides are due via e-mail (ece649-staff@ece.cmu.edu) to course staff:

Must be in PowerPoint (.ppt compatible with Office 2003) or Acrobat (.pdf)

Due Friday evening October 24 at 5 PM

We recommend you get them done before the break, but it's up to you

Presentations will run from professor's laptop to reduce changeover time

Presentation schedule has been posted on course web site since start of semester

We expect you to be there to present with your group

If you need to change slots, it is up to you to find an acceptable swap arrangement

No Recitation 8 on the 17th

- **Recitation slides are online, you should read them on your own/with your group.**
- **Project 8 is due the Thursday after break, so be ready.**

Project 8 Overview

Project 8 is due Thursday after mid semester break

Part 1: more acceptance testing of mid-semester project

You should do this for project 7, but you shall do it for project 8

Part 2: updated scenarios, sequence diagrams, reqs for Fast, Smart elevator

R-T6: The Car shall only stop at Floors for which there are pending calls.

R-T7: The Car shall only open Doors at Hallways for which there are pending calls.

R-T8: The Car Lanterns shall be used in a way that does not confuse passengers. This means:
R-T9: The Drive shall be commanded to fast speed to the maximum degree practicable.

R-T10: For each stop at a floor, at least one door reversal shall have occurred before doors are commanded to Nudge

Part 3: more monitors

Introducing some Safety (Invariant) Monitors: more about this in the recitation
8 slides online

Questions?