Recitation #10

18-649 Distributed Embedded Systems

TA: Rohit Vijayaraghavan

8th November 2013

Electrical & Computer ENGINEERING

Carnegie Mellon

Note: Course slides shamelessly stolen from lecture

Announcements and Administrative Stuff

- Project 11 due Thursday Nov 14th
- Project 12 due Monday November 25th
- Presentation slides due Sunday December 1st at 5 PM
- Presentations week: December 2nd 2013 onwards
- Final project due Tues, Dec. 10th.
- 10th December 2013 is hard deadline.

Weekly Progress Update Page

- **Fill these in status reports every week by the deadline**
- **http://www.ece.cmu.edu/~ece649/progress/**
- **Your participation grade *heavily* depends on these reports**
  - Participation is 5% of total grade
- **Weekly progress updates due every week Friday 9:00 PM**
- **Everyone submits one report each week**
  - Even if they' re late, we still want them (Standard late penalties apply)
- **All students should be able to access the progress page**

A Few Words on Traceability

- **We noticed a few discrepancies in presentations over past few weeks**
  - Some sequence diagrams, requirements, state charts, code, etc. didn't seem to trace correctly
  - If we point out issues during the presentation, make sure you go through your design and look for more similar issues

- **Just as a heads-up, the final project grading criteria requires complete end-to-end traceability**
  - Avoid taking shortcuts with process
  - Introduces errors in design traceability and makes bugs harder to track down
  - End up generating extra work for yourselves

- **You should NOT be using the Future Expansion column anymore to complete your traceability tables.**

Drive Controller Requirements

- **Some question on which requirements take priority**
  - Drive ***should*** be Stopped whenever mEmergencyBrake is activated
  - The commanded value of Drive ***shall*** either be the same as or "adjacent to" the value of DriveSpeed

- **Technically, in simulation, its unclear if it makes a difference**
  - Once the emergency brake is triggered, the simulation ends

- **According to the requirements, adjacency takes priority over the safety brake**
  - *Shall vs. should*
  - This means your Drive has to be designed to sequence Fast ➔ Slow ➔ Stop during an emergency brake event

Final Presentation

1. **Showcase design aspects of your *elevator***
   - You spent the whole semester working on it
   - Tell us about the coolest parts or biggest challenges!

2. **Lessons learned about process**
   - Now that you've had a chance to do a relatively large design project using process, tell us about it
   - Good vs. bad
   - What bugs you found in various phases of review and testing

- **We want to emphasize that there is much more flexibility for content in the design explanation portion than previous presentation**
  - If you're unsure whether what you want to present is appropriate in content or scope, ask us!
  - But, required elements need to all be there (especially the metrics)

Project 11

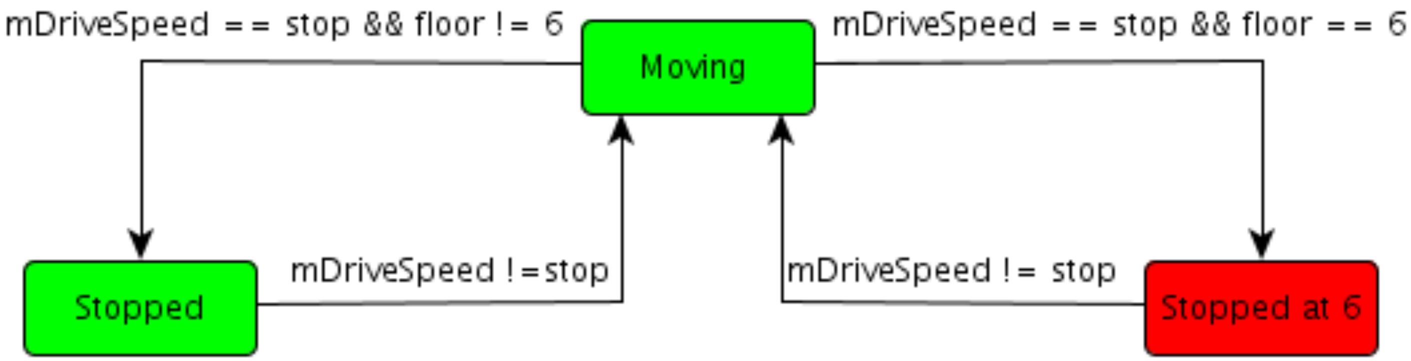
- **Use runtime monitoring to verify high level requirements**
  - Verify R-T6 through R-T10
  - State Chart required for each requirement

- **Implement an advanced runtime monitor**
  - Build upon your project 7 monitor
  - Use this to find requirements violations in your design
    - These violations may not be obvious during acceptance tests
    - Its possible to deliver all the passengers and still violate high level reqs

- **When we grade your project, we run *our* runtime monitor**
  - Don't write yours to handle weird edge cases you know exist in your design
    - Run straightforward tests based directly on the requirements
  - Be thorough! Final Project is worth a big percentage of your grade!

Requirement State Chart Example

- **High Level Requirement: “The elevator shall never stop at floor six”**
- **State charts should:**
  - Mirror the actual state of the elevator
  - Contain both valid and invalid states
  - Throw a warning in invalid states


mDriveSpeed == stop && floor != 6
Moving
mDriveSpeed == stop && floor == 6
Stopped
mDriveSpeed !=stop
mDriveSpeed != stop
Stopped at 6

The monitor is NOT a new controller

- **Monitor takes mostly physical payloads (few network messages)**

- **receive( ) function executes when the physical payload is sent**

```
public void receive(DriveSpeedPayload msg) {
   checkFastSpeed(msg);
}

private void checkFastSpeed(DriveSpeedPayload msg) {
   // Update variables and check for violations
   // If between floors, at some point must go faster than slow speed
   // If reach a new floor and haven't, then print violation
```

- **Monitor must use SystemTimer objects (if you need them)**
  - Don't use Timer objects (only use these in your controllers)
  - This prevents the runtime monitor from contributing to randomness in simulation

Looking ahead to Project 12

- **Introduce faster speed**
  - Commit point can now potentially be multiple floors away
  - May require updating calculation of commit point
    - Depends on your implementation
  - Use "-fs 5.0" to set fast speed to 5 m/s
- **All unit tests must pass**
- **All integration tests must pass**
- **Run acceptance tests**
  - Acceptance tests must run, but do not have to pass
  - Use –b 200 and -fs 5.0
  - If you successfully run at 200k bps or below you get full credit.
- **Update traceability**

Course Project Exit Criteria

- **Run Time Monitor Must Be Implemented**
  - Pass all unit tests with zero failed assertions
  - Pass all integration tests with zero failed assertions

- **Pass all acceptance tests**
  - Using -b 200 and -fs 5.0
  - Zero failed assertions (after startup)

- **Must have a working elevator to complete the course**
  - "Working" means passes the set of tests listed on the final project web page
  - Non-working results in Incomplete if you don't get it working by grade deadline

- **+1% final grade for best elevator (one group only)**
  - Rank groups by average performance and satisfaction across acceptance tests

- **+2% final grade for complete and consistent design portfolio**
  - All groups are eligible for this

Questions?