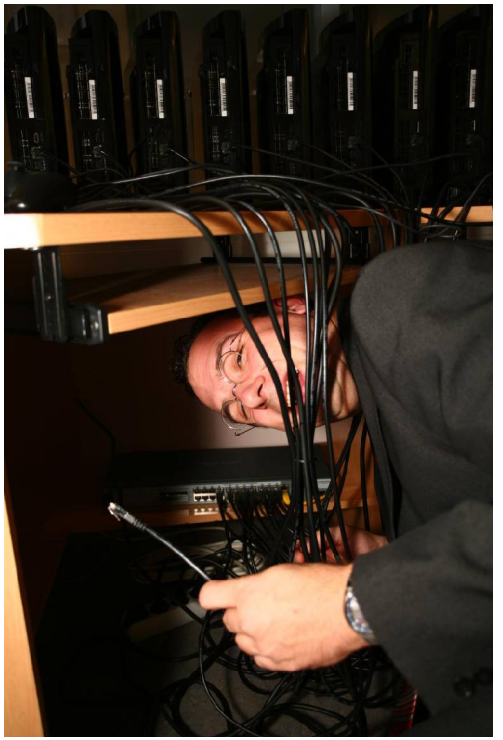
Systems Modeling

Lecture
2009/02/12

Me



- http://ulno.net, J.Liivi 2, room 311
- email: replace http:// with ulno@
- Postdoctoral Researcher at University of Tartu
- PhD from RWTH Aachen University, Germany,Computer Science Department 3: Configuring eHome Systems
- Research: ps3 computing, f2f computing, graph search, educational computer games

This lecture will be interactive!

- Questions desired
- Handouts
- In class work
- Public reports
- Course Project

- Participation has impact on grade
  => Name tags please

Course Structure

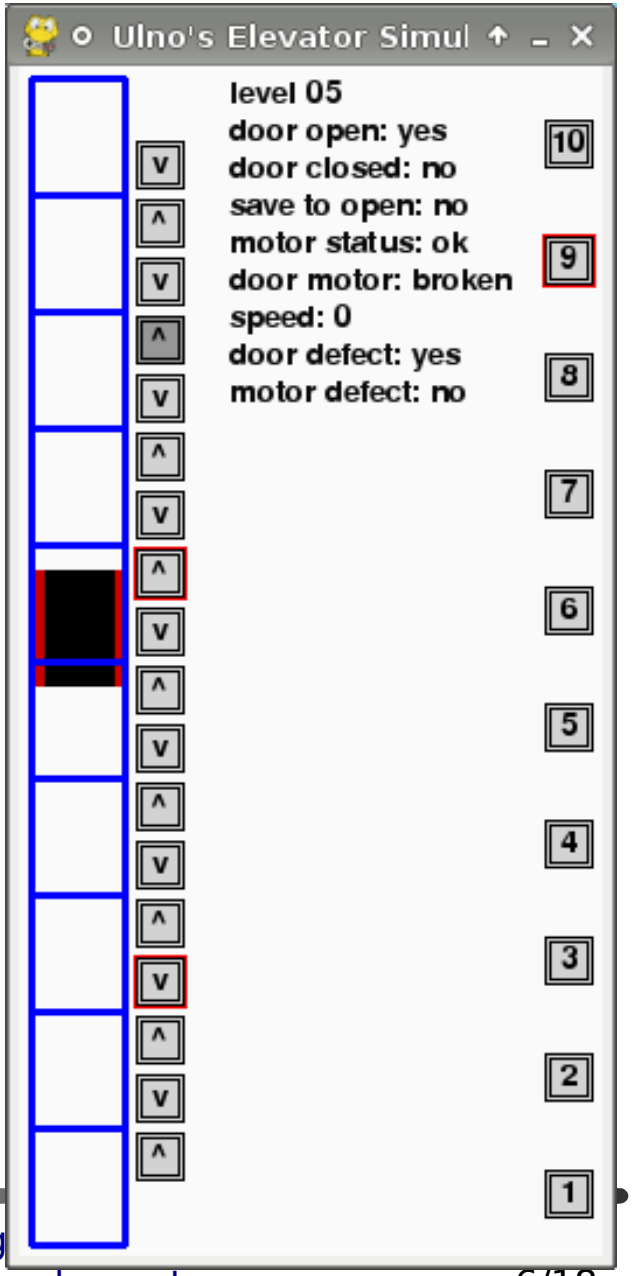
- Effort: 4AP => approx. 160h of work
- Max points for course: 100
- Grades: 100-91:A, 90-81:B, 80-71:C, 70-61:D, 60:-51:E, 50-0:F
- Points distribution:
  - Participation in 12 lectures: 15 points (2 points per lecture possible)
  - Course project: 20 points
  - Exam: 65 points

Schedule

- 12.2. lec 10:15-12:00 404 No.: Introduction, Abstraction and Concretion, Models
- 16.2. lab 16:15-18:00 004 No.: User Stories and Scenarios
- 19.2. lec 10:15-12:00 404 No.: Objects Firsts, Usecases, Object Diagrams
- 23.2. lec 16:15-18:00 004 La.: Requirements Engineering with the Customer
- 26.2. lec 10:15-12:00 404 No.: Class Diagrams
- 02.3. lab 16:15-18:00 004 No.: Argo-UML, Object-, Class-Diagrams
- 05.3. lec 10:15-12:00 404 No.: Statechart and Sequence Diagrams
- 09.3. lab 16:15-18:00 004 No.: Statechart and Sequence Diagrams
- 12.3. lec 10:15-12:00 404 No.: Agile Software Development
- 15.3.-20.3. daily lecture/lab 08:00-12:00 205, Software Design Patterns
- 23.3. lab 16:15-18:00 004 No.: TBA
- 26.3. lec 10:15-12:00 404 Ma.: Petrinets
- 30.3. lab 16:15-18:00 004 Ma.: Petrinets
- 02.4. lec 10:15-12:00 404 Ma.: Petrinets
- 06.4. lab 16:15-18:00 004 Ma.: Petrinets
- 09.4. lab 10:15-12:00 404, Dr. Norbisrath, Final Presentations
- maybe one more session on 13.4. for presentations

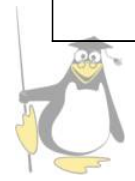
Course Project

- Elevator Control: control the simulator from http://elsim.ulno.net
- Implement in groups with each 3-4 members
- Design and agile development driven approach with ArgoUml



Course Project Assessment

Criteria	Standards				
	Poor (1 point)	**Lacking** (2 points)	**Pass** (3 points)	**Good** (4 points)	**Outstanding** (5 points)
Functionality & Correctness [5 points]	The system runs but is trivial in functionality or full of bugs.		Provides a reasonable level of functionality and relatively few bugs.		Provides an outstanding level of functionality and no obvious defects.
System Design [5 points]	Userstories/ scenarios, object diagrams, usecases, architectural overview are incomplete and faulty.		Userstories/ scenarios, object diagrams, usecases, architectural overview are provided but they are incomplete or the design violates design principles.		Userstories/ scenarios, object diagrams, usecases, architectural overview are provided and sound, interesting, and complete.
Report [5 points]	Report is clearly incomplete		Report has most or all of the required elements		Report has all the elements and it is well-presented.
Demo & presentation [5 points]	The demo did not work properly, presentation was very poor		The demo worked but oral presentation was poor or answers to questions were not satisfactory		The demo fully worked and the oral presentation was clear and convincing



Interactive work

- Write down (10 minutes):
  1) What major problems have you encountered during past software projects?
  2) Estimate what percentage of your time you spend on analysis, design, coding, and testing/debugging/fixing.
  3) How do you estimate, how much effort a project will require?

Why model?

- *“If you were supposed to understand it, we wouldn't call it code.”* -
  from a Federal Expresspromotion, reported by IS Survivalist Matthew Persico
- Enables better communication
- Shows connections, relations, context at once
- Easier to see and estimate risks/costs

Concrete vs. Abstract

- Form teams of 3-4
- Discuss and write down examples for concrete and abstract things (10 examples).
- Create definitions of concrete and abstract.
- Write down two ideas, what this could have to do with modeling.
- Work on this for 15 minutes.
- Report!

Homework

- Form teams of 4 (maybe 3), define a day and time outside my course, where you meet regularly
- Give a list of the abilities/skills and one weakness of every team member concerning a software development project, add email address
- Write one abstract why this combination of abilities makes a good developer team.
- Send all information to my email until evening Feb. 15th (1p of project report)