Model-Based Testing

(DIT848 / DAT261)

Spring 2017

Gerardo Schneider
Dept. of Computer Science and Engineering
Chalmers | University of Gothenburg

gerardo@cse.gu.se
http://www.cse.chalmers.se/~gersch/

Model-Based Testing

- What is testing?
  - The process of systematically experimenting with an object in order to establish its quality
  - Object “=“ software -> software testing
- Why software testing?
  - Most used technique in industry to increase confidence in Sw quality
  - Job possibilities ☺
- What is model-based testing?
  - Generate tests (semi-)automatically from the model of the system under test
- Why model-based testing?
  - Cost saving, systematic approach to testing, automated traceability, early detection of flaws, etc.

Overview course content

- Overview on verification and validation
- Testing in general
  - Black box testing (Junit)
  - White box testing (Coverage analysis)
- FSM / EFSM
- Model-based testing
  - How to select your tests
  - Graph theory in MBT
  - ModelJUnit
  - Making your tests executably

How much of each topic?

- We will discuss today

Guest lectures:

- TBD

Theory and practice

Learning Outcomes

- Explain the distinction between software verification and validation;
- Describe the connection between software development phases and kinds of testing;
- Describe and explain (a number of) different test methods, and use them in practical situations;
- Describe and explain what model-based testing is;
- Construct models in the modeling and specification languages learned in the course;
- Construct appropriate and meaningful test cases, and interpret and explain (to stakeholders) the results of the application of such test cases (using appropriate tools) to practical examples;
- Apply model-based testing on realistic examples;
- Exemplify and describe tools for testing software, and use them and interpret their output;
- Identify and hypothesize about sources of program failures, and reflect on how to better verify the correctness of such programs.

Staff

- Gerardo Schneider - gerardo@cse.gu.se
- Course assistants
  - Per Hallgren – hallgrep@chalmers.se
  - Jan Schröder - jan.schroder@gu.se

Student representatives

- MD. ZUBAYER ALAM (zubayer@student.chalmers.se)
- ZIWEI HUANG (ziwei@student.chalmers.se)
- JAKOB NOETZEL (noetzel@student.chalmers.se)
- RAMI SALEM (srami@student.chalmers.se)

- Need a volunteer from GU:

Course organization

- 12 lectures + tutorials
  - Approx. 2h each, in modules of around 45 min
  - Lectures given at the beginning of the course
  - Tutorials to help on the practical aspects
- Mini-projects – 3 HEC
  - **Mandatory!**
  - Meeting with the assistant on predefined dates
- All the information on the course page
  - http://www.cse.chalmers.se/edu/year/2017/course/DAT261/
  - Considered official! (Any message will be written in the News section under Home)
    - Reason: Students from Chalmers and GU – different systems
- Register to the course Google Group: https://groups.google.com/forum/#!forum/dat261-model-based-testing-2017
- Individual written exam – 4.5 HEC

Literature

- M. Utting and B. Legeard, **Practical Model-Based Testing**. Elsevier (Morgan Kaufmann Publishers, 2007)
  - An electronic version is available at http://bit.ly/wGIT94 (you must be logged in Chalmers' network to get access)
- Other interesting books
  - P.C. Jorgensen. **Software Testing: A Craftsman's Approach** (Auerbach Publications, 3rd edition, 2008)
  - Sommerville…
- Other books and references
  - See list in course homepage

Mini-Projects

- Check course homepage: “Project Assignment” tab
- Each group will be assigned a part of a SUT (an editor)
- Groups of 3 students
- Important Dates:
  - **Wed Mar 22**: Deadline Forming groups
  - Wed Mar 22: Distribution of the SUT
  - **Mon Apr 5**: Deadline Assignment 1: Testing using Junit
  - **Mon Apr 26**: Deadline Assignment 2: Modeling (EFSM)
  - **Wed May 3**: Deadline Assignment 3: ModelJUnit and IntelliJ
  - **Wed May 15**: Deadline Assignment 4: Implementing your model
  - **Wed May 17**: Final Presentation Mini-Project
  - **Mon May 22**: Deadline final report
- To pass you need to pass both the mini-project and the written exam
- Information about the mini-project, how to proceed to “register” your group, etc, is posted in the homepage
- Register to the Fire system (information in the webpage)
- Any question or doubt contact the course assistants

Written Exam (Individual)

- Written exam: May 31, 2017 at 8h30 (Johanneberg)
- Re-exam: August 21, 2017 at 14h00 (Johanneberg)

Important!

- The exam is designed to increase the confidence that a student passing the course achieve the Intended Learning Outcomes
- **Strongly recommended to learn when you work on the assignments!**
- So, *most probably* the exam will consist in 3 tasks widely covering the content of the course
- Example (3 tasks, 20 points each): You will need to have at least **30**/60 points for getting **G (3)** (and at least **40** points for **4**), **and at least a minimum of correct answers for each task** (e.g., **5 points** per task for 3, **7 points** per task for 4)
- To get **VG (5)** you will need to have at least **50**/60 points **and at least a minimum of correct answers for each task** (e.g., **12 points** per task)
- *Open book* exam modality

Important: About examination!

Open book exam**:** You can bring one book and one piece of paper with personal notes (no electronic devices nor other material)

For GU students registered in 2015 or later, and for Chalmers students registered in 2016 (DAT261):

- There are 2 ”moments”
  - Individual written exam: 4.5 HEC – U, G (3,4), VG (5)
  - Assignment (mini-project): 3.0 HEC (U, G)

For GU students registered before 2015 and Chalmers students registered before 2016 (DAT260):

  - Individual written exam: 7.5 HEC – U, G (3,4), VG (5)
  - Or talk with me in case you want to have a short exam and do the mini-project

Changes w.r.t. last year
Result of course evaluation

- Give more support at the beginning with IntelliJ
  - Assistants will help students to set IntelliJ
- Keep the SUT (System Under Test - miniproject) but do not change the parts to be analyzed by each group
  - Last year each group got a part of IntelliJ at the beginning and then change it (getting a new one with errors). Not so now!
- Continue putting announcement in the News of the course page, but repeat them in Google group too
- Lectures at the beginning was good
  - We keep that
- From previous years: More about modeling (EFSM)

What is your background?

- Programme
  - Software Engineering: 18
  - Other: Communication Engineering: 1,   CS: 2
- University
  - Chalmers: 19,   GU: 2
- Knowledge on logic?
  - Propositional (classical): 8 (FOL: 2)
  - Other: 1 (LTL)
- Which imperative/OO prog. lang. do you know?
  - Java: ALL
  - C (C++): 14
  - Other: 4
- Which functional prog. lang. do you know? (for information): 8
  - Haskell: XX
  - Erlang: XX
  - Other: XX
- Knowledge on Testing? 2
- Knowledge on automata theory or FSM (Finite State Machines)? 11

Preliminary schedule

- Is the content "appropriate" according to your background?
- Remember requirements for the course:
  - General programming knowledge in *imperative/object oriented* (ideally Java)
  - Knowledge of *propositional logic*
  - Have some experience in *testing/debugging* your own programs

Wish list…

- What are your expectations?
- Something you would like to learn on testing (or verification in general) not covered in the programme?
- Are there topics you already know and don't want to see again?

About Registration…

- If you are a **GU student**
  - You need to register through the *Student portal* at GU
- For **Chalmers** students (late registration)
  - Contact the *Studieexpeditionen* (student office) student_office.cse@chalmers.se

Questions?

- Register to the course Google Group: https://groups.google.com/forum/#!forum/dat261-model-based-testing-2017
- Register to the Fire system: Check the webpage

Check the course page regularly

Hope you enjoy the course!