

Model-Based Testing

(DIT848 / DAT260)

Spring 2013

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
- FSM / EFSM
 - (More this year after evaluation!)
- Black box testing
 - JUnit
- White box testing
 - Coverage analysis
- Model-based testing
 - ModelJUnit
- Property-Based testing
 - QuickCheck

How much of each topic?

- We will discuss today

Guest lectures:

- QuickCheck

- ...

Learning Outcomes

- Describe 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
- Guest Lecturers (To Be Confirmed)
 - **John Hughes** (Chalmers)
 - **Thomas Arts** (QuviQ)
- Course assistant
 - **Hamid Ebadi** - hamide@chalmers.se

Student representatives

- TBA

Course organization

- 15 lectures (approx. 2h), including guest lectures
- 7 assignments
 - Meeting with the assistant on a weekly basis
- All the information on the course page
 - <http://www.cse.chalmers.se/edu/year/2013/course/DAT260/>
 - Considered official! (Any message will be written in the **News** section under Home)
- Written exam

Schedule lectures

- “Theoretical” lectures on **Mon** and **Wed morning**
- Exercise supervision on **Wed afternoon**

- The *Monday* slot is 08.15-12.00
 - Lectures will in general be approx. 2hs
 - Any wish on starting later? (**9.00**, **9.15**?)

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...
- Papers on QuickCheck
 - See course homepage
- Other books and references
 - See list in course homepage

Weekly assignments

- Given every Wednesday (except holidays; and one Monday)
 - Assignment related to the week lectures
- Not mandatory, but... **strongly recommended to work on the assignments!**
- The assistant will give you feedback on your assignments if you ask for
 - You will get information on how to submit
 - The assistant will let you know about submission "deadlines"
 - If everybody decides to submit it might be required that you work on groups
- Take feedback seriously
 - It's part of your learning
- If recurrent questions/problems with assignments - discussion during the lectures
 - If you have questions/doubts about the exercises be sure you ask the assistant during the consultation meetings

Exam

- Written exam: June 1st, 2013
- Re-exam: August 26, 2013

Important!

- The exam is designed to increase the confidence that a student passing the course achieve the Intended Learning Outcomes
- **Strongly recommended to work on the assignments!**
- *So, most probably* the exam will consist in 5 tasks widely covering the content of the course
- You will need to have at least 50/100 points for getting **G (3)** (and at least 65 points for **4**) and at least a minimum of correct answers for each task (e.g., 8 points per task)
- To get **VG (5)** you will need to have at least 80/100 points and at least a minimum of correct answers for each task (e.g., 12 points per task)
- *Open book exam modality*

What is your background?

Number of students: 21 + (at least 8 GU students)

- Knowledge on logic?
 - Propositional (classical): 5
 - Other: -
- Which functional prog. lang. do you know?
 - Haskell: 6
 - Erlang: 3
 - Other: 1 (Lisp)
- Which imperative/OO prog. lang. do you know?
 - Java: 15
 - C (C++): 17
 - Other:
- Knowledge on Testing? 6 (3 took Testing, debuggin and verification)
- Knowledge on automata theory or FSM (Finite State Machines)? 3
- Knowledge on QuickCheck? 2

Preliminary schedule

- Is the content “appropriate” according to your background?
- Remember requirements for the course:
 - General programming knowledge in both *imperative/object oriented* and *functional* programming
 - 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?

Questions?

Check the course page regularly

Hope you enjoy the course!