

# Model-Based Testing

(DIT848 / DAT260)

Spring 2012

**Gerardo Schneider**

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

[gersch@chalmers.se](mailto:gersch@chalmers.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
- 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
- Black box testing
  - JUnit
- White box testing
  - Coverage analysis
- Model-based testing
  - ModelJUnit
- Property-Based testing
  - QuickCheck

## Guest lectures:

- MDE and test
- QuickCheck
- Test generation from KeY

Theory and practice

# 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** - [gersch@chalmers.se](mailto:gersch@chalmers.se)
- **Guest Lecturers (To Be Confirmed)**
  - **Christos Dimou** (Univ. Carlos III, Madrid)
  - **Gabriele Paganelli** (Chalmers)
  - **Thomas Arts** (QuviQ)
- **Course assistants**
  - **Hamid Ebadi** - [hamide@student.chalmers.se](mailto:hamide@student.chalmers.se) (1st part)
  - **Pablo Buiras** - [buiras@chalmers.se](mailto:buiras@chalmers.se) (QuickCheck)

# Student representatives

- OLOFSSON, ANDERS (oloande@student.chalmers.se)
- MELLGREN, VIKTOR (mviktor@student.chalmers.se)
- RAIHLE, JACOB (raihle@student.chalmers.se)
- BACKLUND, EMIL (backlune@student.chalmers.se)
- ZHU, XI (xizh@student.chalmers.se)

# Course organization

- 15 lectures (2h), including guest lectures
- 7 assignments
  - Meeting with assistants one hour per assignment
- All the information on the course page
  - <http://www.cse.chalmers.se/edu/year/2012/course/DIT848/>
  - Considered official! (and information in these slides takes precedence)
- Written exam

# 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 Thursday (except holidays)
  - Assignment related to the week lectures
- Not mandatory, but... **strongly recommended to work on the assignments!**
- The assistants will give you feedback on your assignments if you ask for
  - You will get information from them on how to submit
  - Assistants will let you know about submission "deadlines"
  - If everybody decides to submit they might require 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 assistants during the consultation meetings

# Exam

- Written exam: **May 21, 2012**
- Re-exam: **August 30, 2012**

## **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*

- Which functional prog. lang. do you know?
  - Haskell:
  - Erlang:
  - Other:
  
- Which imperative/OO prog. lang. do you know?
  - Java:
  - C (C++):
  - Other:

Questions?

Check the course page regularly

Hope you enjoy the course!