Module folder for CS 6214 – Automated Software Validation

Abhik Roychoudhury
School of Computing
National University of Singapore
abhik@comp.nus.edu.sg

1 Background of the course

CS 6214 is an advanced graduate level course focusing on formal software validation. In short, this course covers different techniques to give functionality *guarantees* about software. Developing reliable software is a difficult task which most Computer Science professionals have to face. Furthermore, reliability guarantees are particularly important for safety critical software. This course familiarizes our graduate students with some techniques which can help in this regard. The course is very timely with the increased role of software in controlling devices of everyday use (popularly known as embedded systems).

2 Intended Audience and Learning Outcomes

1. This course will help graduate students (particularly those who are interested in Programming Languages and Software Engineering) to be familiar with some of the ongoing cutting-edge research topics in program validation and verification. This will be allow students to

   - identify future research topics
   - apply part of the knowledge in their respective research areas.

   Indeed this learning outcome has achieved in the past years, with many School of Computing Ph.D. students working in related areas familiarizing themselves with formal techniques via this course.

2. Many of the techniques to be discussed in this course have reached some amount of industrial maturity e.g. model checking. The students can employ this knowledge later in an industrial setting e.g. knowledge of model checking is very useful in the Electronic Design Automation (EDA) industry. This is further possible since each technique is explained with concrete tools (e.g. the discussion on model checking is accompanied with detailed lectures on the SPIN model checker.

3 Assessment Scheme for CS 6214

- *Term Project*: This is usually done individually and it accounts for about 30-35% of the grade (slightly varies from one offering of the course to another). It involves either surveying an important cutting edge area in software validation or modeling and validation a substantial piece of software/protocol. In each offering of the course, one of these modes is tried out to ensure a fair assessment for all students (*i.e.* either all students are engaged in survey or all are doing implementation projects).
- *Midterm Examination* Written, Open-book Examination held in the 7th week of the semester. It accounts for about 20% of the grade.
- *Final Examination* Written, Open-book Examination. It accounts for about 45-50% of the grade (slightly varies from one offering of the course to another).

4 Information about CS 6214 Term Projects

I would like to illustrate with a concrete example the activities involved in a CS 6214 project and the activities the students go through in course of the term project.

4.1 Schedule

The rough schedule for the projects are as follows.

- *At the end of 4 weeks*, the students submit a 2-page project proposal with details of deliverables by mid-term and end of term.
- *At the end of 8 weeks*, the students submit a midterm project report discussing the progress, the work left and any difficulties they faced so far. At this point of time, I also meet up with the students to monitor their progress.
- *At the end of the semester*, the students submit a project report and make a 15 - 20 minutes presentation (along with questions/answers). The presentations are attended by all the students, so the students can ask each other questions during the presentation.

4.2 Sample Projects

The key topics covered in CS 6214 are as follows.

- Model Extraction (Abstraction of programs to models)
- Model Checking (Proving properties of Models by Search)
- Theorem Proving or Deduction

In the following, I provide the details of some sample projects ¡undertaken by the students of CS 6214 in the 2004-05 offering. All the projects in this offering were implementation-oriented projects. I have listed how the topics covered in the course are tried and tested in more practical settings via the projects. Also, there is sufficient diversity and spread among the projects even though they are all concerned with practical software validation.

- *Designing and validating an elevator control system*
  The tasks of this project involved programming the control system in C, extracting a model from the program (using the Modex tool) and then using the SPIN model checking tool for formal verification.

- *Set sharing analysis of CHR programs*
  The tasks of this project involved programming language level abstraction and implementing a customized program analyzer in a constraint programming framework.

- *Use of theorem provers for checking properties of web ontologies expressed in OWL*
  This project involved formalization of the problem in first-order logic and theorem proving using the PVS prover.

- *Possibility of using model checking for WCET analysis of embedded software*
  This project investigates a cutting-edge research area: whether model checking can be used for bounding the execution time estimates for programs. The student was involved in investigating the practicality of the approach by conducting experiments in the SPIN model checker.

- *Visualizing output of model checking and connecting it to source-code*
  For practical debugging, interpretation of the model checking output and connecting it with the program's source code is very important. This project involved making changes inside the Bandera toolkit so that forward and backward associations between models and programs can be maintained.

5 Student Profile

The students attending this course are typically Ph.D. students or students intending to do Ph.D.

6 Contact Hours

Apart from the lecture hours, I meet up with the students regularly to discuss the projects. In the later offerings of the course, this has been done more systematically by creating study groups of two students. This forms an affinity group for discussion of course material and projects. The meetings can then also proceed more systematically. Without the affinity group, in the past students have come to me with various kinds of problems – even problems in debugging their code when they have been stuck for a long time. The formation of the affinity group can ease these issues with the students learning from each other more. This is also suited in a small class setting, where a mechanism like "Discussion Forum" may not be necessary.

7 Use of Information Technology

Throughout the course, the Integrated Virtual Learning Environment has been extensively used. All lectures have been posted via IVLE. In particular, the Lesson Plan feature of IVLE has been extensively used to structure the course. This allows me to highlight to the student the compulsory and additional readings in every lecture. The Lesson Plan for CS 6214 has been attached.

The E-reserves facility of NUS libraries has also been used to post selected chapters of books. There is no single recommended textbook for the entire course.

8 Lecture Material and Examinations

The course/lecture structure can be seen from the IVLE lesson plan which has been attached in this module folder. In addition, the following materials are also attached.

- Sample Lecture on Theoretical materials covered in the course
- Sample Lecture on Practical Tools discussed in the course
- Sample Final Examination (2004-05).
- Sample Midterm Examination (2004-05).

In the sample theory lecture, I have marked how I give examples/exercises to make the concepts clear and how I try to remove common misconceptions (see header page of the lecture).

In the sample tool lecture, I have marked how I familiarize the students with the tool usage through interactive sessions (see header page of the lecture).

In the sample exams, I have marked how the students have been asked a combination of questions involving problem solving, technical depth and application of knowledge in new settings (see header page).

9 Improvements based on Student Feedback

Based on the student feedback from the last two year (2003-04 and 2004-05), the following improvements have been made to CS 6214.

- *More orientation of the course towards practical side* More attention have been given towards software validation tools. In particular, almost 3 lectures have been devoted on orienting the students towards two popular validation tools – the SPIN model checker and PVS theorem prover.

- *Re-organization of lecture material* To address this comment, I have tried to re-organize the flow of the course. Previously, I used to discuss the generation of models from program code at the very beginning. Even though logically this is the right thing to do before moving on to model checking, the students found it a bit hard. This is because they lacked detailed understanding of the models so they found it difficult to visualize the transformation of code to models at the very beginning. To prevent this disruption, I now focus on modeling and model checking first before discussing code → model transformation (please refer to the Lesson Plan for details).

- *Providing break in the lecture* This is now being provided.

10 Appendices in CS6214 Module Folder

1. Lesson Plan for the module from IVLE.

2. Sample Lectures on Theoretical materials covered in the course

3. Sample Lecture on Practical Tools discussed in the course

4. Sample Midterm Examination (2004-05).

5. Sample Final Examination (2004-05).

6. Sample Term Project Report from 2004-05 (this project involved use of theorem proving using the PVS tool).