SDM: summary, revision, exam

Perdita Stevens

School of Informatics
University of Edinburgh

Learning objectives

1. Design simple object-oriented systems.
2. Create, read and modify UML diagrams documenting designs, both on paper and in an appropriate tool.
3. Determine whether a UML model and a body of Java code are consistent; if they are inconsistent, identify the inconsistencies precisely and propose remedies.
4. Evaluate and evolve object-oriented software designs, making use of common design patterns if appropriate.
5. Discuss the use of modelling and model-driven development tools in software development, e.g. why and how models of software can have varying degrees of formality, capabilities and limitations of the tools.

Using SDM

- Please remember what you learned here when you develop software in future – e.g., in your project. This doesn't mean "draw UML diagrams for everything" (think about whether they're useful in the circumstances!) but it does mean "make your designs SOLID"!
- Please do well in your exams...

SDM exam

Three questions, each worth 25 marks:

- COMPULSORY QUESTION 1
- you choose between question 2 and question 3.

I aim to balance:

- bookwork
- straightforward application of knowledge
- more challenging problem-solving.

Everyone examined this semester (December 19th: but official Registry page is definitive).

Example exam question types

Be prepared to answer exam qustions *such as*

- You are to design a system to do .... Draw a ... diagram to illustrate ....
- Explain what ... means in this ... diagram/OCL/xtext
- Write OCL/xtext to express ...
- Write an essay on ... drawing on [specific required reading; OR, anything you know]
- Imagine you are ... Suggest how you should use [modelling; DSMs; patterns; design techniques; MDD tools] to help you work.
- To make the following design decision, what information would you need and why?
- What is the ... design principle/pattern? Apply it/explain whether it is useful in this situation ...

General approach to SDM exams

Many questions lend themselves to addendums like "Comment briefly on [missing or unclear requirements, design alternatives, problems in a design]" – this is your chance to show that you can think like a designer! Don't woffle. Do show understanding. (The mark scheme will typically say something like "1 mark each for any two reasonable points, e.g ...")

Write clearly and concisely.

Assume your papers are being read by a reasonable – but picky! – human being:

If you are confused about what a question wants you to assume, briefly say why, say what you are assuming, and do what seems best to you.

Writing implements

If you pay close attention to detail, you will notice that the script books and a page on the university site says you have to do exams in pen, not pencil.

This has caused concern for exams with diagrams in – are you really supposed to draw them in pen?

Two things are true:

1. It looks as though the University really does say Yes. So do.
2. But personally I can't imagine any Board of Examiners in Informatics agreeing to dock marks from someone who wrote a correct answer in pencil!

Do not waste time redrawing a whole diagram if you make an error. If you leave your script in any state where I can tell that you meant the correct thing, you will get the marks.

Examinable material

Basically, everything mentioned on the schedule page that isn't specifically excluded, e.g.:

- lectures
- labs (using common sense: I would not expect you to explain in detail how to make Papyrus do ...; I do expect you to know what the tools used are for and how they are applied, and, e.g., to write simple OCL constraints and xtext rules);
- required readings (using common sense: I'm not testing your photographic memory, but I expect you to have synthesised the knowledge and to be able to apply it);
- patterns exercise sheets (see list of examinable patterns on slides);
- videos, including both the ones I made and the others

Revision resources

Feel free to post any questions to Piazza.

Do complete the labs if you didn't!

Sample exam paper. (We didn't do ATL in the end though!)

Recent (2013 on) SEOC exam papers (on university past paper site): somewhat useful but syllabus of SDM is pretty differerent.

Please fill in the questionnaire!

Would very much appreciate feedback on your experience of this course – it will likely not be presented next year, so I'll have time to act on it before the next presentation!

Key question I'd particularly like input on this year:

- What did you think of the labs?
- If you didn't turn up to labs and/or lectures (and are reading this slide online), why not? Was it a problem (they were inaccessible or not useful?) or were you happily doing labs remotely and watching lecture videos?

I am always interested to hear from past students now working in software development. Please drop me an email some day!