

Guidance for CS405 exam 2008

CS405 will be a 3 hour examination. There are 5 questions. Question 1 [20 marks] is compulsory. You select any 2 from the remaining 4 [each accounted 15 marks]. This means an approximate division of time: $72 + 2 \cdot 54 = 180$ minutes in total. It is probably a good idea to tackle Question 1 first - it has been known for students to get so carried away in answering the optional questions that they leave insufficient time for the compulsory question.

Question 1 is about the fundamental principles and concepts of "modelling with definitive scripts" (MWDS). You are expected to understand how MWDS develops and exploits the concepts of *an observable*, *a dependency* and *an agent* and how these relate to the notion of "making a construal". It is useful to be familiar with the variety of ways in which observables, dependency and agency are represented in EM models. Important also to appreciate the essential empirical rather than formal nature of these concepts. Essential to have reflected on how these concepts fit into the context of MWDS "vs" conventional programming. For instance, you should be familiar with the shift in perspective that is discussed and motivated in Lectures 7-11, which compare and contrast MWDS and "definitive programming" with functional programming, programming-with-dependency, and with spreadsheet development and use. Being familiar with some of the key diagrams, and knowing how to use these in an explanatory way in conjunction with standard models is valuable. More advanced development of these issues can be found in Lecture 14 (Rethinking Programming). The discussion of possible applications for MWDS illustrated in seminar 1 in connection with Sudoku solving is also relevant and informative. You should also be able to draw on your experience of modelling for your WEB-EM-04 submission.

Because of the large amount of material that was distributed during the module, and the absence of external resources, some specific guidance on the topics that feature in questions 2-5 follows.

Overall the four optional questions represent different perspectives on the module. Two are rather practically oriented, one towards construction of models and the other towards comprehension. The remaining two questions are more reflective in nature. In one of these the emphasis is on ideas developed within EM and presented in the module for which no in-depth study of external sources is required. In the other, familiarity with external sources is essential.

The practical questions will be primarily based on the laboratory work in Labs 1-3 but draw on knowledge of two models that have been a particular focus for attention in the module. These are "the OXO laboratory" that was the theme of Lab 3 and the 'heapsort' model that was studied in some detail in the Lecture 14 (Rethinking Programming). A useful foundation for answering the practical questions can be gained by looking at the sample questions on the labsheets in conjunction with studying the heapsort model. Note that to appreciate the practice fully and so give good answers to the practically-oriented questions, you will still need to revise the module quite broadly, and will be expected to be familiar (e.g.) with the principles of giving an LSD account (cf. Seminars 2 and 3) and the use of the dependency modelling tool (cf. Lab 3).

Of the reflective questions, one addresses a specific broad topic that was central to the module (viz. the application of EM to concurrent systems modelling), the other is concerned with EM in its relation to established thinking about computer science. If you wish to tackle these questions in the examination, you are strongly advised to plan and draft essays in advance of the examination.

For the application of EM to concurrent systems modelling, you will need to study lectures 15-18 inclusive, and ensure that you are familiar with the principles for interpreting LSD accounts using the ADM that are illustrated in a model such as the railway station animation and were discussed in Seminars 2 and 3. The eight lectures under the **Concurrency** tab on the left-hand panel of the webpage are useful supplementary material, but only the subset of these lectures and associated material that is directly referenced in lectures 15-18 and seminars 2 and 3 are required reading.

The discussion of EM as a possible alternative framework for thinking about computing was a central theme of Lecture 13. The references cited in that lecture (Brooks, Harel, Cantwell-Smith, Ridley) are good sources for motivating a shift in perspective on computer science theory in its relation to software development, knowledge representation and databases. There is also useful relevant material in Lecture 1, Lecture 14 and lectures 19 and 20.

This guidance should be helpful in guiding and focusing your revision. Though specific lectures have been highlighted in connection with particular questions, it is advisable to have a general familiarity with the material in

the module, as there is a lot of interconnection between the various themes. It's also important to be familiar with a variety of standard models, and with some of EM papers (especially those highlighted in the lectures). Credit will be given for references to models for illustration and to papers where appropriate.