

Revision guidance for the CS405 examination 2009

This is the exam rubrick:

CS4050 Introduction to Empirical Modelling
Time allowed: 3 hours
Answer question 1, and two other questions.
Question 1 carries 20 marks. The other questions carry 15 marks.
Credit will be given for evidence of familiarity with
a variety of standard illustrative models.

Pay particular attention to "familiarity with a variety of standard illustrative models", as this makes communication much simpler. It is often easier to explain an issue with reference to a specific model than to discuss it in the abstract. Models you can refer to include: the room / roomviewer, heapsort, jugs, graphics presentation, cruise control, railway animation, digital watch, Clayton Tunnel. A small number of standard models (sometimes somewhat simplified) is readily accessible via the recently added Web EDEN webpage cited at the foot of the CS405 module webpage.

To assist your revision for the examination, I shall indicate the main themes (but not the detailed content!) of each of the five questions on the paper, and indicate which lectures etc are most relevant in each case. It is of course desirable that you have a good overall grasp of the module, as this will be helpful in answering all questions.

Here is an index to the themes of the exam questions:

1. Artefacts, Agentification and Animation as they apply to modelling a railway accident scenario [20]
2. Definitive notations: design and implementation [15]
3. The OXO laboratory [15]
4. EM and constructionism [15]
5. EM in relation to traditional Computer Science [15]

It is a good idea to have a specific choice of questions in mind prior to the examination, and to focus your preparation on this. There are broadly two kinds of ways in which your revision can be informed - by practical studies (drawn from the case-studies discussed in the lectures, the labs and seminars, and your own personal experience of model-building), and by the more theoretical/philosophical /methodological reflections on the nature of EM that have been presented throughout the module. All the questions require something of both kinds of knowledge, but the balance is different in each case. In broad terms, questions 2 to 5 progress from a more practical to a more theoretical emphasis.

Revision notes question by question

Question 1: This question gives you the opportunity to show that you have understood how EM applies to making **construals** in the context of studying "commonsense concurrency". The principal lectures relating to this can be found via the Concurrency link on the left-hand panel of the EM webpage. (There is a useful overview of the what is discussed in all eight lectures at this link.) Lectures 1, 2 and 3 set the scene for discussing artefacts, agentification and animation, and these ideas are developed in lectures 5, 6, 7 and 8. You will be expected to show understanding of how EM artefacts are used in representing agent interaction and of the roles of LSD and the ADM. For illustrative purposes, you will be expected to sketch how you might apply these concepts to modelling a simple railway accident scenario. This is not one of the scenarios that was explicitly discussed in the

lectures (Tamworth, Abermule, Clayton Tunnel), but of course will be amenable to similar kinds of analysis (see Lecture 17 and Seminar 2 for background). Basically, you will be given an informal description of an accident and asked to illustrate how you would use EM principles and tools to make an appropriate construal. Credit will be given for showing familiarity with key concepts. In EM, it is not generally appropriate to give a formal definition of a term (see the discussions in the lectures of terms such as 'construal', 'observable', 'dependency', 'agent', 'oracle', 'handle' etc), but good to talk around each of these terms and give illustrative examples based on standard models and on the scenario in the given case-study.

NB: Question 1 is compulsory, and carries 20 marks. It is probably a good thing to tackle it first - given that the exam is 3 hours in total, it should occupy you for about 72 minutes. You have about 54 minutes to devote to each of the two questions you choose from questions 2 through 5.

Question 2: This question is concerned with *definitive notations*. You'll need to show that you know

- what a definitive notation is (w.r.t. notion of underlying algebra, and a syntax for defining observables and dependencies),
- that you're familiar with different examples (e.g. Donald, Scout, Sasami, Eddi, Gel, Arca, Cadence and the definitive subset of Eden itself),
- how a definitive notation is typically implemented in the hybrid environment of Eden,
- how the agent-oriented parser can also be used to specify definitive notations (e.g. the implementation of Eddi).

You should also be familiar with key meta-issues that relate to definitive notations: the problems of specifying dependencies when the observables are complex (consider e.g. shapes/openshapes in Donald, problematic aspects of defining lists / list components in Eden, moding in Arca); the pros and cons of having a diversity of notations. You'll get credit for being able to give examples of the use of definitive notations (for instance, Eddi is the subject of Lab 4 from 2007-8 - see the online notes on Seminar 1, and Gel is used in graphicspresHarfield2007).

Good sources for this question are Labsheets 1 and 2, and the additional reference material associated with these (including the online notes for Seminar 1).

Question 3: This question takes "The OXO Laboratory" as its theme. This term is introduced in "EM for the Single Agent" - the second lecture in the series of eight devoted to Concurrency. It relates to the model `oxoJoy1994` and to a number of variants that you can find elsewhere (e.g. `oxoGardner1999` and the online Web EDEN mirrorOXO). The construction and character of these models was discussed in Lectures 1 and 2, and (as indicated in the online notes on Seminar 1) is also the theme of Lab 3 from 2007-8. The OXO model was also the theme of a comparison between functional programming and EM that was the topic of Lecture 12. The OXO Laboratory was also the basis for introducing William James's notion of "radical empiricism". The online paper "Experimenting with Computing" referenced in Lectures 2 and 12 is another useful source.

Question 4: This question focuses on EM for educational technology, with specific reference to constructionism. This theme that was introduced in connection with the discussion of Logo in Lecture 10, was amplified in the discussion of teaching heapsort in Lecture 11, and was also the principal subject of Lecture 20. To get a good focus on this question you need to be familiar with Papert's notion of constructionism, and have a good grasp of how EM models can be used in an educational role. In addition to the specific references associated with Lecture 20 (which include an extract from Chris Roe's PhD thesis that gives some essential background), you will need to revisit the discussion of EM in relation to basic Logo and Imagine-Logo as presented in Lecture 10. Some key concepts are: the distinctive nature of the support that EM gives to the roles of teacher, pupil and developer (all of whom interact with one and the same "state" through making redefinitions, but in such a way as to reflect their different skills and agendas), and the Experiential Framework for Learning (see Lecture 21). There are many other related resources on EM and educational technology accessible via the "Application areas / Educational Technology" link on the left-hand side of the EM webpage. If you do this question, it is important to be aware of some of educational applications of EM - the

account of how Eddi was once applied in teaching relational databases in year 2 (see link from online notes on Seminar 1) is one practical example. An online Web EDEN bubblesort model may also be useful.

Question 5: This question relates to the claim that EM is associated with a broader view of CS. This claim was set out in the material that was presented in the first lecture of the module - Steve Russ's powerpoint slides "Empirical Modelling: a broader view of computing" and the associated 'alternative curriculum' for computing science set out in "An EM perspective on computing". As a result of attending / studying the module, you should by now have a better grasp of the nature of that claim, and what issues may be problematic. The principal module content is sandwiched between Lecture 1, which sets out the claim, and Lectures 21 and 22, which show that other commentators (in a variety of areas of computer science, such as software engineering, AI and databases) also have deep concerns about the traditional foundations of computer science, and hint at how these might be addressed in an EM framework. To tackle this question, you should do some background reading on the writings of Brooks, Harel, Cantwell-Smith and Ridley. You should also give some thought to how the curriculum set out in Lecture 1 might be developed in the light of the rest of the module.