CS405 Introduction to Empirical Modelling 18/11/03

The practical exercise for CS405 will be complemented by a written examination in which there are two sections. The first section will contain a single compulsory question, and the second a choice of two from four questions. The material for these questions will be drawn from the themes represented in the 6 groups of EM-related questions listed below. To have sufficient background for the examination, you should expect to have studied about 4 of these 6 groups of questions in some detail.

Some of the themes below have already been referenced in previous lectures. The remaining 9 sessions for CS405 will be devoted to seminar-style presentations that point to EM papers and models together with other references for you to consult.

1. Definitive notations

Should notations be unified?
What is the future for definitive notations e.g. will they proliferate?
In what sense is a definitive script an artefact?

What principles govern the design of a good definitive notation?
How general-purpose / domain-specific / application-specific should definitive notations be?

To what extent is a functional programming environment a general-purpose definitive notation?

2. Systems Development / Concurrent Engineering

What role can EM play in requirements modelling and domain understanding?
What potential does EM to support distributed development?

To what extent can EM be regarded as a system development method?

In comparison to traditional software development approaches:
- What is the role of dependency in software development?
- Will EM modelling methods scale-up?
- Is object-orientation is essential ... ?
- Can / how does EM address conceptual integrity?
- What connections does EM have to 4GL programming?
- What role is played by circumscription of behaviour?
- What is the scope for translation?

3. Implementing EM

In what sense is the ADM ... a machine? ... an observation regime?
What is the relationship between LSD accounts and LSD specifications?
Why hasn't LSD got an operational semantics?

What are the principal deficiencies of tkeden?

Can we implement dependency maintenance at the low-level? .. in hardware?

How are agency and dependency linked?

How to implement dependency maintenance that embraces privileges, values and references?

4. Education

To what extent is traditional computer programming well-aligned to domain learning?

Does classical programming support bricolage-style development? ... situated modelling?

What are the qualities of spreadsheets for educational environments?

Are / how are the agendas of the educationalist and computer specialist linked?

Can EM be developed as a viable end-user modelling paradigm for teachers / pupils?

5. Human Computing

What is meant by the method-user-tool framework? In what respects can EM be seen as offering an alternative framework?

What do you understand by Human Computing? Where does word-processing fit in?

What technological trends motivate the study of Human Computing?

How is EM oriented with respect to HCI, ease-of-use and invisible computing?

6. Empirical Modelling and Radical Empiricism

What is meant by saying that "one experience knows another"?
What are the motivating ideas behind seeking non-logicist foundations for AI?
In what respects are these similar to James's motivations in RE?
... and to the motivating concepts behind EM?

Can EM be given a mathematical foundation?
What marriage - if any - is possible between logic and EM?
In particular, can EM be related to formal methods?
... and if not, can EM be used in safety critical applications?