

## Seminar F4: Future directions and projects

Return to the opening slide for M4 Seminar:

areas Programming Paradigms, Applications, Foundations, SE

### Software Engineering

- Propose a software development method. Work so far is oriented towards reactive systems; projects currently addressing data intensive applications. [Many existing programming techniques developing along these lines in the UIMS, data-base and spreadsheet culture.]

Concerns to connect with formal specification. Connections with Z etc still hazily understood. Also origins of LSD in SDL. Would be good to evaluate in comparison with Lotos, SDL and Estelle. Other work related in the real-time and concurrency area: French school, StateCharts, UNITY, (Numerical) Petri Nets. Relationship to protocol specification and testing.

### Programming Paradigm

- Expectations in the abstract programming arena different from those predominantly considered in this course. Our approach doesn't put the emphasis on algorithmic issues in the first instance. Is state-based though, and can be applied to modelling real/abstract machine models. In particular, intend to explore its relevance to portability concerns and to parallel machine models. Prototyping from definitive specifications is of particular interest here. Potential re general-purpose programming to be considered.

Relevance to complexity theory, where there is a significant lack of appropriate ways to measure performance formally. Want to count abstract operations generally, so declarative languages are useless here, but procedural languages do not necessarily have the right expressive power. Not enough to construct abstract data types and calculate numbers of abstract operations. Of particular interest are complexity measures that concern maintenance of dynamic data structures. Possible interest in exploring definitive models in that area.

Examples of interesting challenges for definitive programming:

1) write an LR parser in such a way that the finite state machine associated with the grammar is specified by a definitive script.

2) write a text editor (developing Edward Yung's Unconventional Text Editor)

3) write definitive implementations of definitive software tools

## AI and Applications

Design (especially for engineering) is an area of particular interest.

Neural networks.

Semantics of semantics of programs: form vs content. Reference information.

## Foundations

How to express "all"! notions of CS in terms of semantics we are developing for LSD and definitive notations. Examples of concepts we can deal with to some extent:

agent, process, object, observation, experiment, program, execution.

More work needed on synchronisation, data types (specify the stack etc), time, constraints, hypermedia ...

## Potential MSc projects involving parallelism:

1. Railway Signalling (MSJ)
2. Extending the ADM
3. Parallel implementation of EDEN
4. Parallel implementation of ADM
5. Comparative study of concurrent systems modelling techniques  
e.g. SDL vs Agent-oriented modelling, Statecharts vs LSD
6. Extension of LSD design:  
parametrisation, roles and subagents, agent references
7. Code generation for parallel machine from definitive specification
8. Definitive implementation of EDEN
9. From requirements to formal specification of reactive systems
10. Formal association of requirements and testing
11. Applications in scientific visualisation with parallelism (cf GRN)
12. A definitive notation to specify synchronisation of events
13. Communications network simulation (BT)
14. Development of an environment for modelling reactive systems
15. Comprehensive formal analysis of a small LSD specification
16. Elaboration of LSD concepts in respect of  
intention, obligation of agents  
agents, roles, objects, processes etc

Proposals for joint projects may be considered, subject to approval of GRM.

*Not so much directly parallel in orientation:*

Donald with objects

ARCA revisited

Abstract definitive programming