Seminar M4:
The Definitive Programming Project: Motivation / History / Context

M4.0. Issues for contemporary computer science
programming paradigms
software engineering
applications and AI
foundations

M4.1. History of the Definitive Programming Project

Sequence of topics considered

Relationship to parallelism and concurrency

M4.2. Programming paradigms: motivating ideas

Backus and FP: limitations of FP
history sensitivity etc

tension: good mathematical semantics vs rich expressive power

State-based vs declarative methods

Relevance parallel programming:
general-purpose parallel programming?

Can respectable programming be liberated from a declarative style?

Development of OOP: how POOL has developed etc

Interaction in Miranda

M4.3. Advantages of definitive programming re interaction and design

Where are we? where can we go next? representation of state

Making state respectable: an alternative framework for semantics

Definitions for dialogue and design
e.g. the room through the eyes of the architect, the stored door etc

Virtues for design

Representing undefined values
Easy undo
Design and simulation in parallel

M4.4. General-purpose programs using definitive representations of state

single agent, double agent, multi-agent models

Virtues of state representation for parallelism

Order of definitions immaterial
Conceptual simplicity of parallel redefinition model
Idempotence of definition
Dependency recorded
Deals with undefined values gracefully:
can represent effects of incomplete computational processes

What must be done to represent computation in terms of multi-agent definitive models?

References

Backus
Harel
Booth
Kent
Cantwell-Smith