Systems development and EM

EM for Systems development

'Concurrent system in the mind of the external observer'

- identifying an objective perspective
- circumscribing agency
- Identifying reliable generic patterns of interaction

Concurrent engineering design task …

Concurrent Engineering view

Have a design team. Need to

- represent many alternative views
- distinguish and synthesis knowledge of many different kinds
- deal with concurrency, inconsistency and conflict
- record human decision-making and negotiation
- express the concept of a consensus view

Abstract Definitive Machine

… in principle supplies a very general framework within which to address all these issues

- represent state at all levels of abstraction using definitive scripts
- represent agent interaction at all levels of abstraction as redefinition with scripts
- combine manual and automatic redefinition reflecting interaction by manager, designer, engineer, user

Routine vs creative design

Building a system that can fulfil a specific requirement from machine-like components of proven reliability with identified function and range of application

E.g. sequential programming, object-based design, catalogue-based design

Building an environment within which systems and requirement can be identified: reconciling what we *believe* to be true with what we *observe* to be true

EM as pre-system development

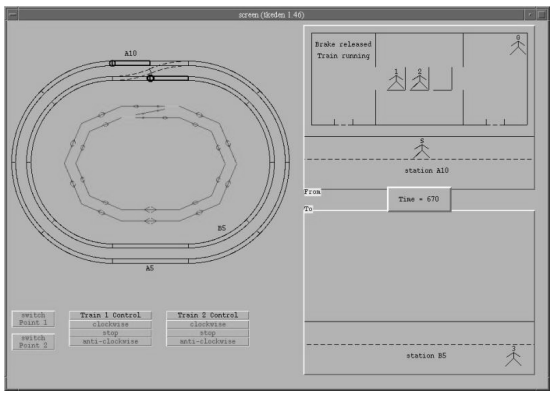
Making the transition from

uncircumsribed ill-conditioned, loosely regulated interactions

to

circumscribed precisely prescribed well-regulated reliable behaviours

See this in railway history …



Issues for development in EM

- negotiation and elaboration
- learning as involved in requirements and design
  "growing software"
- development as situated problem-solving – amethodical software development
- traditional systems/programs derived by circumscription and optimisation
- how far can object / agent abstractions help?

Issues for development in EM

- *negotiation and elaboration*
  cf. experiential framework for learning
- *learning as involved in requirements and design*
  "growing software" – Brooks, WMB+SBR
  Racing Cars – Simon Gardner
- *development as situated problem-solving – amethodical software development*
  car maintenance analogy
  Pi-Hwa Sun PhD Thesis, Paul Ness, Y-C Chen

Issues for development in EM

- *traditional systems/programs derived by circumscription and optimisation*

cf. OXO in Pascal programs
(~wmb/public/projects/games/OXO/PASCAL)
Allan Wong PhD Thesis – 'beyond systems'

- *how far can object / agent abstractions help?*
  original aspirations of object-oriented programming
  Lind – notion of 'very weak agent' - McCarthy

The logicist debate

McDermott – a Critique of Pure Reason

Celebrated renunciation of faith in logic as basis for AI

Responses collected in Comput. Intell. Vol 3 1987

Brian Cantwell Smith – a non-logicist stance …

Two Lessons of Logic

Smith identifies two factors of a symbol system:

"first factor" - form
shapes of the symbols
how put together / taken apart
operations and behaviour

"second factor" – content
what the symbols mean, what they are about

Logical models

Have a logical specification (e.g. in the form of a set of predicates formulated over a formal logic)

Associate with this a 'model': some concrete instantiation of the logical specification, in which the terms in the specification have interpretations and the predicates express valid relationships between them

cf. the relation between a specification and a program

The two lessons

Lesson one: the irreducibility of content to form

content relations aren't computed: how symbols 'reach out and touch someone' - almost total mystery

Lesson two: a single theoretical stance

The two factors are relatively independent, but have to be ultimately related: a single, unified theory must provide an account of both factors

The two lessons reframed

Lesson 1

Logic doesn't give an account of

"why" or "how it is that" a particular model is associated with a given specification

Lesson 2

An account of logic has to be framed with reference to both specifications and models, even though logic encourages us to think of them independently

Formal logic tenets *not* for AI

1. Use can be ignored
2. Locally the two factors can be treated independently, even though must be globally related
3. Language and modelling should be treated completely differently (promiscuous modelling: can substitute model of X for X with theoretical abandon)

Formal logic tenets

1. A formal specification is deemed to express 'everything that is interesting' about any model from some significant perspective – a formal sentence has an objective meaning however it is applied
2. For the purpose in hand, everything of interest about a model can be expressed / addressed abstractly
3. Every detail of the specification is potentially highly significant and discriminates between different models: any models of a specification are equivalent

Formal logic tenets *not* for AI

1. *Use can be ignored*

cf. natural language – context dependence, resolving 'now', pronouns etc

2. *Locally the the two factors can be treated independently, even though must be globally related*

'if the first factor could be cleft from the second factor, would make sense to write things down first and build programs second' (McD) – not like thought

Formal logic tenets *not* for AI

3. Language and modelling should be treated completely differently

Whole new theories of representation and correspondence will be required:

- explaining computational practice
- promiscuous modelling pernicious where fine grained questions are concerned
- representations in current computational systems range continuously from linguistic to virtually iconic

What Then?

[In] the empirical view ... as reality is created temporally day by day, concepts ... can never fitly supersede perception

William James: A Pluralist Universe c.1900

A rationalist critique of William James's position

... mere experience ... furnishes no consistent view. [The direct products of experience] I find that my intellect rejects because they contradict themselves. They offer a complex of diversities conjoined in a way which it feels is not its way and which it can not repeat as its own ... For to be satisfied, my intellect must understand, and it can not understand a congeries [ie an aggregate] in the lump.

F H Bradley: Appearance and Reality

James' counter-view

To be 'conscious' means not simply to be, but to be reported, known, to have awareness of one's being added to that being ... The difficulty of understanding what happens here is .. not a logical difficulty: there is no contradiction involved. It is an ontological difficulty rather. Experiences come on an enormous scale, and if we take them all together, they come in a chaos of incommensurable relations that we can not straighten out. We have to abstract different groups of them, and handle these separately if we are to talk of them at all. But how the experiences ever *get themselves made*, or *why* their characters and relations are just such as appear, we can not begin to understand.

William James: Essays in Radical Empiricism

An engineer's outlook ...

Blind variation ... interaction without complete or adequate guidance, leading to discovery ...

Walter Vincenti

What Engineers Know and How They Know It: Analytical Studies from Aeronautical History, 1993