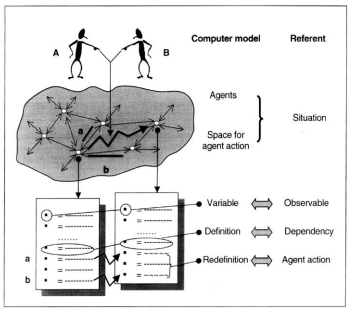


Empirical Modelling for computer-based construals

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 …

EM as pre-system development

Making the transition from

uncircumscribed ill-conditioned, loosely regulated interactions

to

circumscribed precisely prescribed well-regulated reliable behaviours

Illustration from railway history …

… Tamworth 1870 accident

System development in EM

Have shown how the context for the use of LSD and the ADM can resemble that associated with railway history – an empirical evolution from wild to tamed behaviours …

… aim to understand this in its relation to orthodox studies of system development

Empirical Modelling as conceived by WMB + SBR in *Empirical Modelling of Requirements* (1995) …

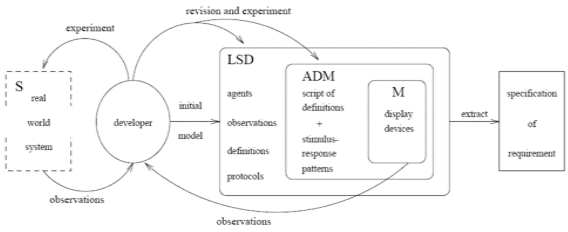


Figure 1: The empirical modelling process
"Empirical Modelling"

Actual development post-conception of WMB + SBR in *Empirical Modelling of Requirements* (1995) …

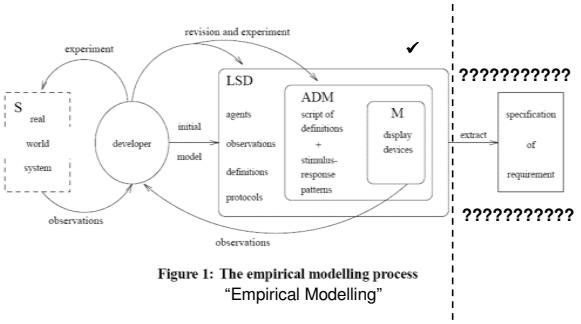
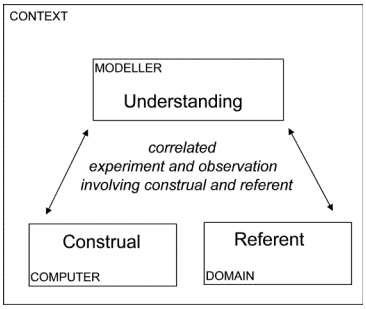


Figure 1: The empirical modelling process
"Empirical Modelling"

Emphasis since 1995 has been largely on modelling with EDEN …



Construal through engagement with the open referent rather than specification through abstraction from the circumscribed referent

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
- what role for object / agent abstractions?

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'
-
- *what role for object / agent abstractions help?*
  original aspirations of object-oriented programming
  Lind – notion of 'very weak agent' - McCarthy

private experience / **empirical** / **concrete**

interaction with artefacts: identification of persistent features and contexts

practical knowledge: correlations between artefacts, acquisition of skills

identification of dependencies and postulation of independent agency

identification of generic patterns of interaction and stimulus-response mechanisms

non-verbal communication through interaction in a common environment

directly situated uses of language

identification of common experience and objective knowledge

symbolic representations and formal languages: public conventions for interpretation

public knowledge / **theoretical** / **formal**

An Experiential Framework for Learning (EFL)

System development in EM in relation to:

1. agent-oriented software engineering
2. non-routine / radical / creative design
3. visual support for reactive system design
4. concurrent engineering aspirations
5. Brooks's notion of *conceptual integrity*
6. principles of HCI: ease-of-use, affordances

1. Agent-oriented software engineering

McCarthy as cited by Lind …

- Jürgen Lind, Issues in Agent-Oriented Software Engineering, LNCS 1957/2001
- John McCarthy, Ascribing Mental Qualities to Machines, in *Philosophical Aspects of AI*, Harvester Press, 1979

2. 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

Normal vs radical design

Michael Jackson, *What Can We Expect From Program Verification?* IEEE Computer, October 2006, 65-71

W.G. Vincenti, *What Engineers Know and How They Know It: Analytical Studies from Aeronautical History*, The Johns Hopkins Univ. Press, 1993.

Michael Jackson (software consultant)
What Can We Expect From Program Verification?
IEEE Computer, October 2006, 65-7

Program verification assumes a formal program specification. In software-intensive systems, such specifications must depend on formalization of the natural, nonformal problem world. This formalization is inevitably imperfect, and poses major difficulties of structure and reasoning. Appropriate verification tools can help address these difficulties and improve system reliability.

Jackson cites Vincenti

W.G. Vincenti distinguishes normal from radical design …

In normal design, "the engineer knows at the outset how the device in question works, what are its customary features, and that, if properly designed along such lines, it has a good likelihood of accomplishing the desired task."

In radical design, by contrast, "how the device should be arranged or even how it works is largely unknown. The designer has never seen such a device before and has no presumption of success. The problem is to design something that will function well enough to warrant further development."

Perspectives on design …

- Primary emphasis of Jackson's paper is on how to tame radical design problems and replace them by normal design problems for which "formalization of the natural, nonformal problem world" is conceivable
- Note that neither an engineer nor an Empirical Modeller seeks such formalization necessarily or characteristically

3. Visual ("experiential") support

- David Harel *On visual formalisms* CACM, 31(5) 1988

  Associated with the invention of the statechart and Harel's stance in *Biting the Silver Bullet*

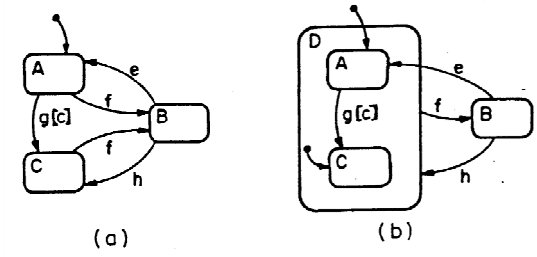
  A style of thinking carried forward in UML and in Harel's work on *Play-in Scenarios*

David Harel's statechart concept

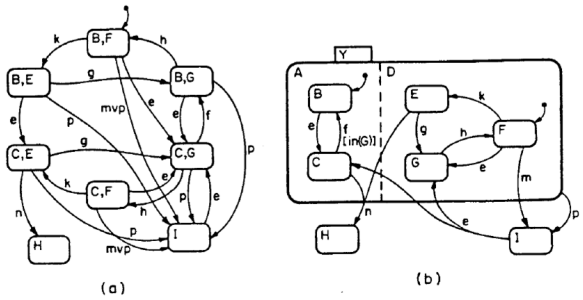
On Visual Formalisms, CACM 31(5), 1988

Generalisation of the FSM diagram …

Statechart = state diagrams
+ depth + orthogonality
+ broadcast communications

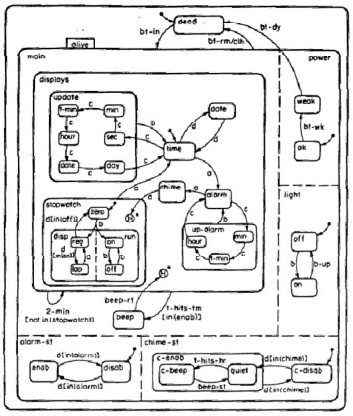


Depth in statecharts



Orthogonality in statecharts

Harel's statechart for the display states of a digital watch



From the conclusion to Harel's paper "On Visual Formalisms" 1988

We are entirely convinced the future is "visual." We believe that in the next few years many more of our daily technical and scientific chores will be carried out visually, and graphical facilities will be far better and cheaper than today's. The languages and approaches we shall be using in doing so will not be merely iconic in nature (e.g., using the picture of a trash can to denote garbage collection), but inherently diagrammatic in a conceptual way, perhaps also three-dimensional and/or animated. They will be designed to encourage visual modes of thinking when tackling systems of ever-increasing complexity, and will exploit and extend the use of our own wonderful visual system in many of our intellectual activities.

4. 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

Extension of Railway Station Animation

- illustrates diversity of the interpretations of redefinition within the ADM framework
- redefinition expressing agency within the model, of the designer, of the user etc
- potentially represents many design views: model railway operation, simulation of station protocol, track layout design (geometry and connectivity), track segment simulation …

Extending the RSA *issues* …

- generalising arrival-departure protocol from one station to more (passenger steps on line when train arrives in other station)
- integrating with the manual operation of trains on track (passenger gets off when train makes an unscheduled stop)
- modeller has to intervene as super-agent to close a door before train will depart

Using the ADM for concurrent design …

A figure taken from EM paper 040

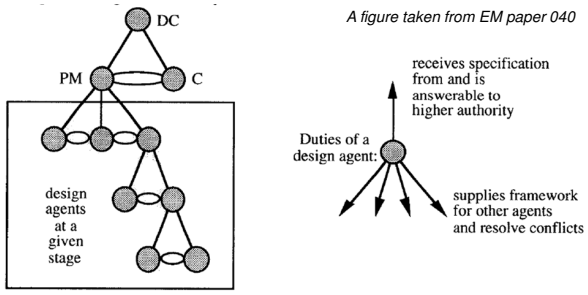


Figure 2: The agent hierarchy and the design participant view

NB The hierarchy is not necessarily static, but typically adapts dynamically

5. Conceptual Integrity

The railway animation model has no conceptual integrity! - but has capacity for concurrently representing many views …

… taking account of situation

accommodating ignorance

capacity to represent nonsense

Slogan: Not *what it represents* but *how it relates to what it doesn't represent*

6. Principles of HCI …

- distinction between development and use is in the mind of the modeller
- *use* as a social construction ("constructivist computing")
- method-tool-user
  - → experiment-instrument-maker
- not primarily concerned with "ease-of-use" – emphasis on engagement cf. construal
- dependencies as generalised affordances
- SEMI aspects of state …

Digital watch model illustrates …

- Chris Roe's created digitalwatchRoe2001 - adapted from digitalwatchFischer1999
- based on his (then) actual digital watch
- can switch between one model and the other by exchanging Roe's informal mode visualisation for the statechart
- has a small set of exercises associated
- EM paper 069 as published in LNAI 2117

SEMI aspects of state …

- situational aspects of state: e.g. using digital watch to time runners
- explicit aspects of state: e.g. details of the appearance of the digital watch
- mental aspects of state: e.g. knowing what mode the watch is in at any time
- internal aspects of state: e.g. what is the power level in the battery