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 …

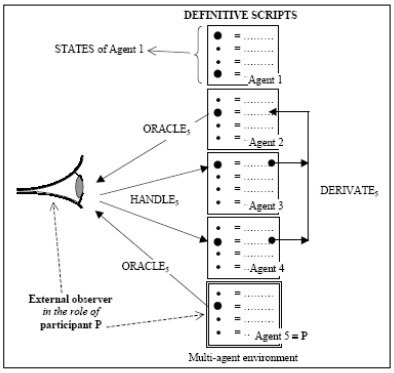


Figure 2-19: Definitive script as observer's model of state ('multi-agent' modelling)

EM as pre-system development

Making the transition from

uncircumscribed ill-conditioned, loosely regulated interactions

to

circumscribed precisely prescribed well-regulated reliable behaviours

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?

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

Software failure scenarios

- Plugger incident
- 3 Mile Island
- North America power outage
- Radiation therapy software failure
- Problem frames – e.g. relating state of lift, state of control software in case of failure

- all connected with sw in its broad context

Software in context …

- Change in context => change in software
- Aspire to model the dependency between sw and context cf. "separation of concerns"
- Key problem – the frame problem
- Significance of construal & prior experience
- Determining the agents of change then rationalising and restricting their agency

Virtues of EM construal …

- Maintaining link between sw & context
- cf. solving the chessboard jigsaw puzzle by turning it upside-down
- Not infallible, in much the same way that there is no ultimate defence against natural disaster
- But taking prior experience into account in our construals helps to address problems

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*

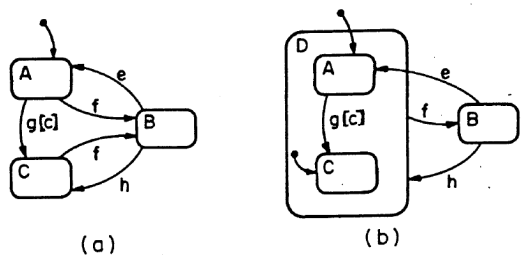
  A style of thinking carried forward in UML and in Harel's work on *Play-in Scenarios*

David Harel's statechart concept

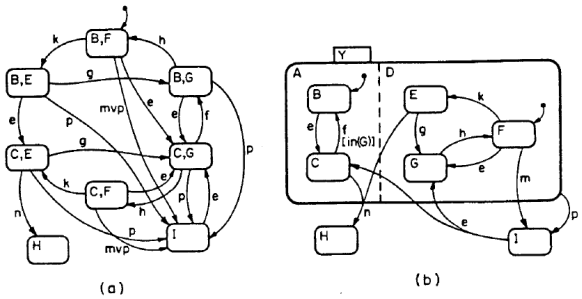
On Visual Formalisms, CACM 31(5), 1988

Generalisation of the FSM diagram …

Statechart = state diagrams
+ depth + orthogonality
+ broadcast communications

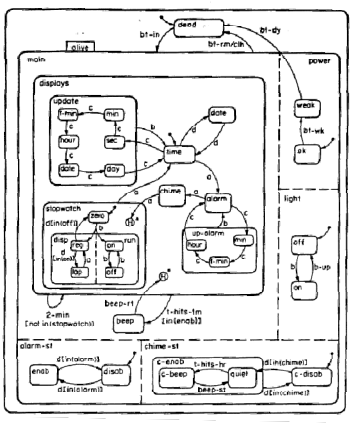


Depth in statecharts



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.