

The value in muddling around modelling

Marian Petre



The Open
University

Centre for
Research in Computing



Doug Lennox



The Open
University

Centre for
Research in Computing



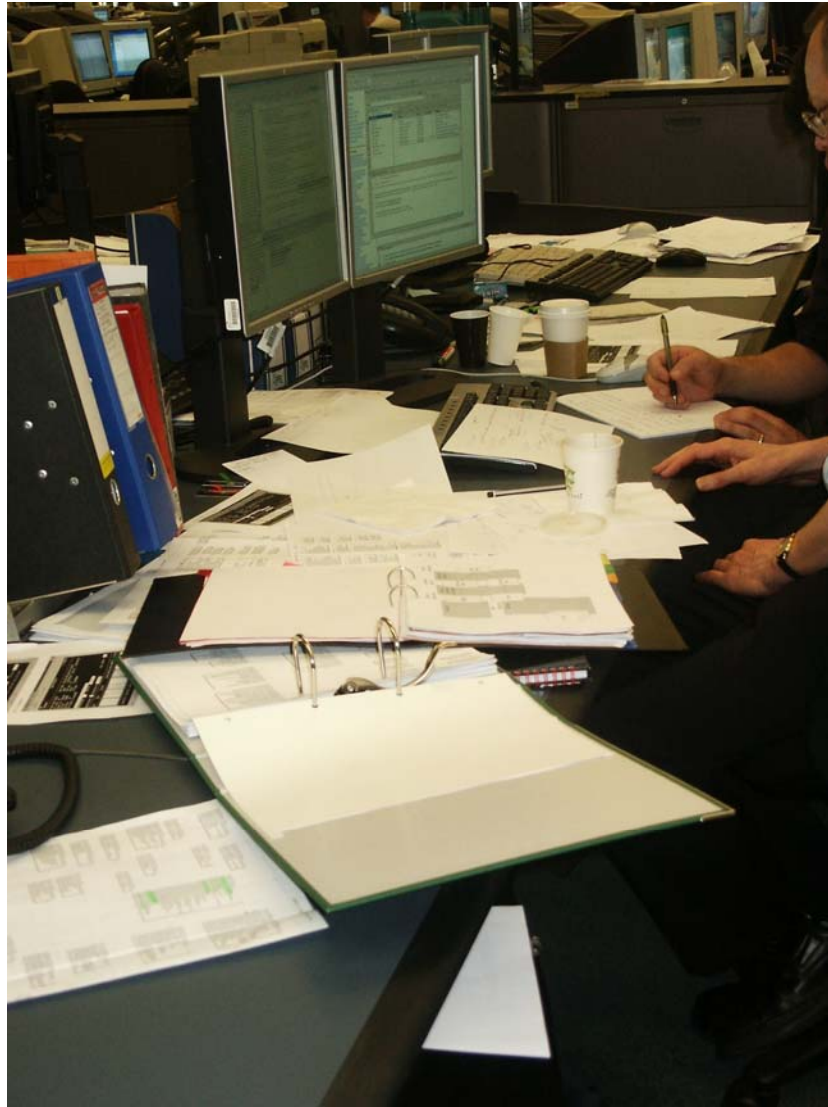
Helen Sharp

Kevin Lynch



The Open
University

Centre for
Research in Computing



Helen Sharp

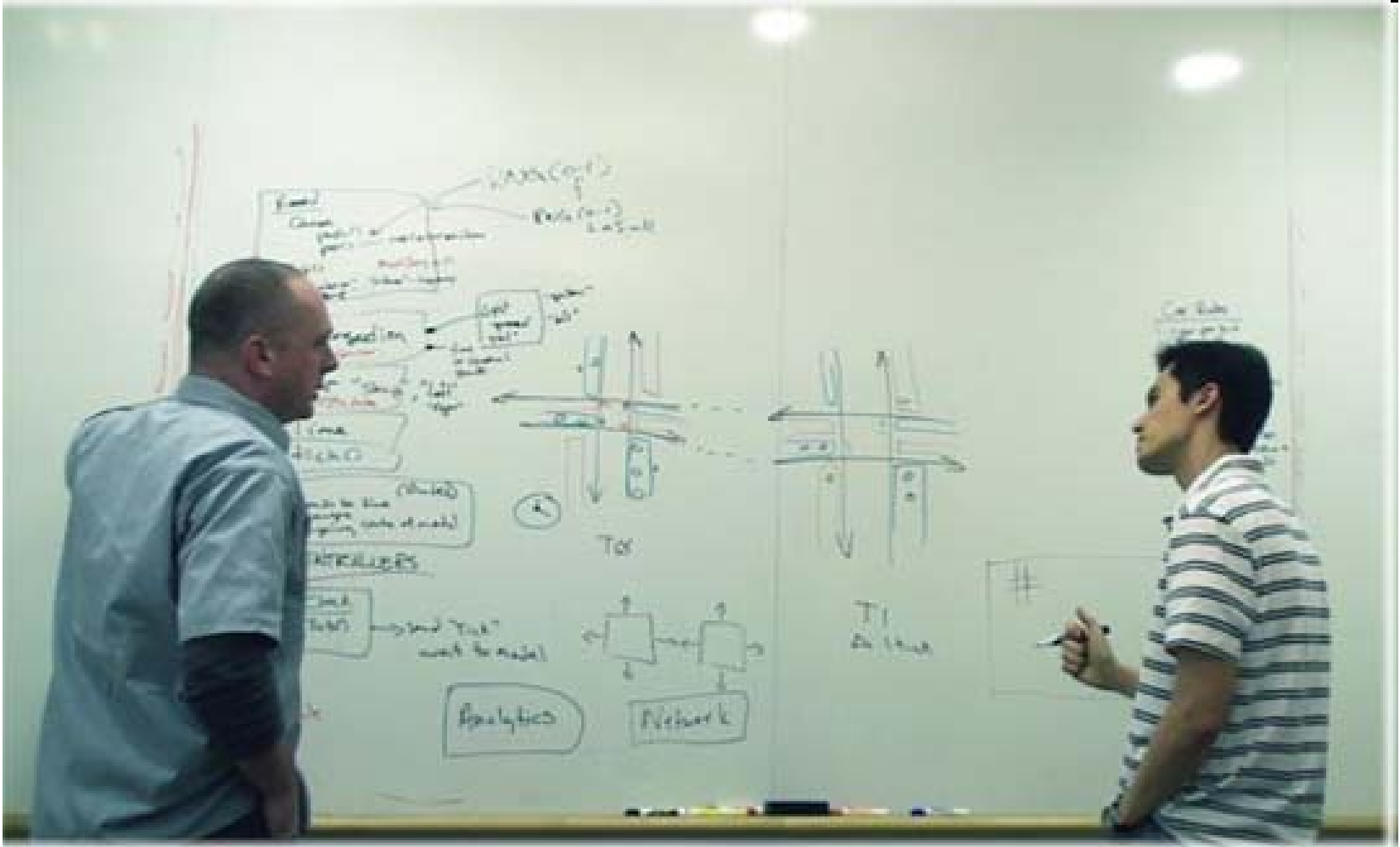


Hugh Robinson



The Open University

Centre for
Research in Computing



Studying Professional Software Design workshop



The Open University

Centre for
Research in Computing



Max Pate – Reflection – Te Papa

- exploration
- consideration of alternatives
- domain : software
- representation
- human reasoning



something about...

- 1.what I've seen
- 2.what's in designers' minds?
- 3.representations of design?
- 4.reasoning about hard problems?
- 5.expert design behaviour?



2010

– Flexible modelling tools

– How experts reason about software: design and problem solving strategies

2005

– Representations for early capture of design ideas

– Disciplines of innovation; mechanisms of inspiration

2000

– Communication and coordination of design ideas within teams

– Software visualisation and mental imagery

1995

– Cognitive dimensions of notations

– Graphical and textual notations

1990

– Programming paradigms



Autodesk

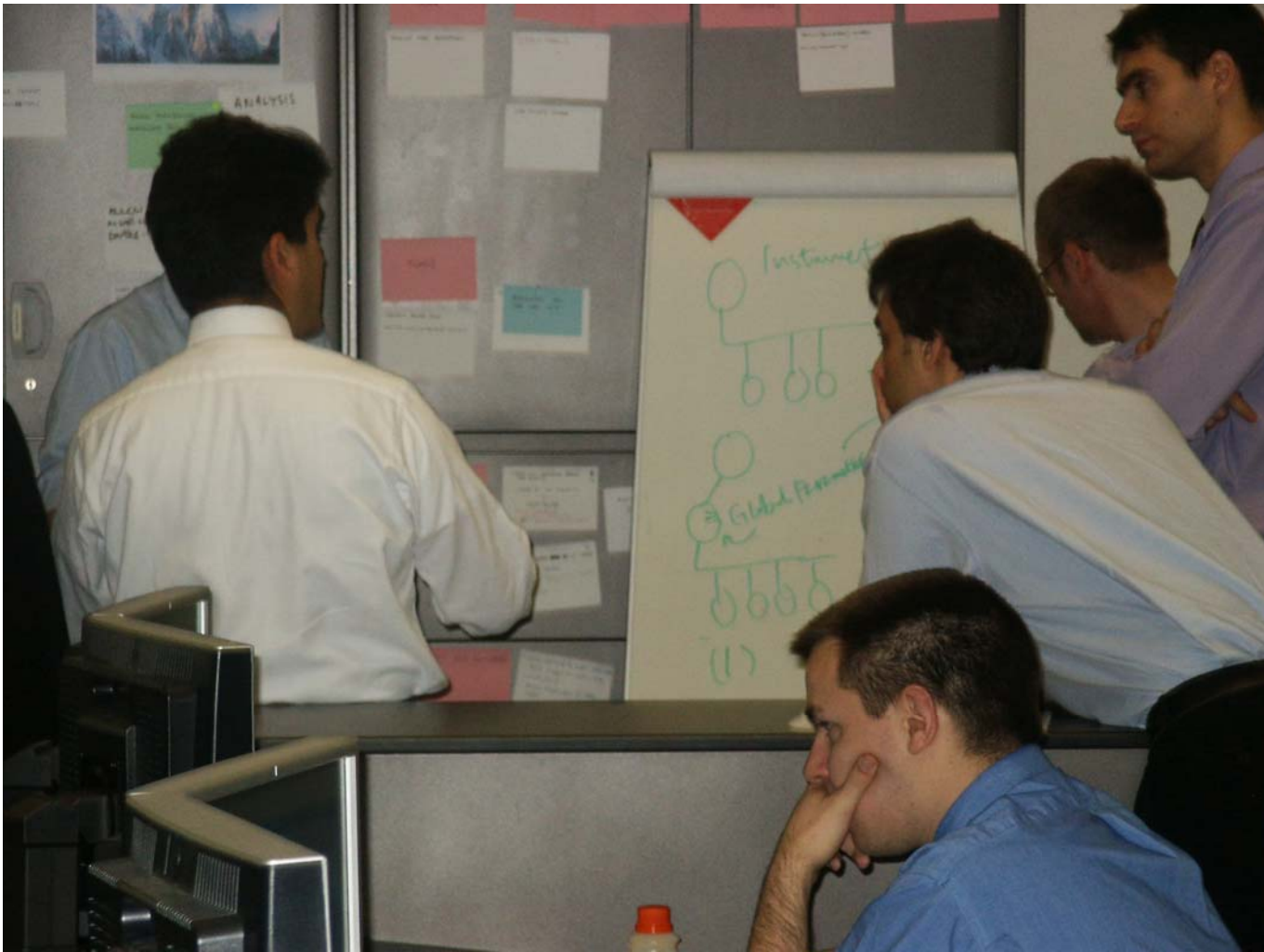


The Open
University

Centre for
Research in Computing

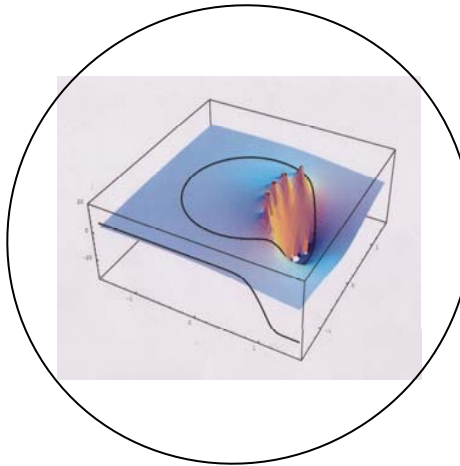




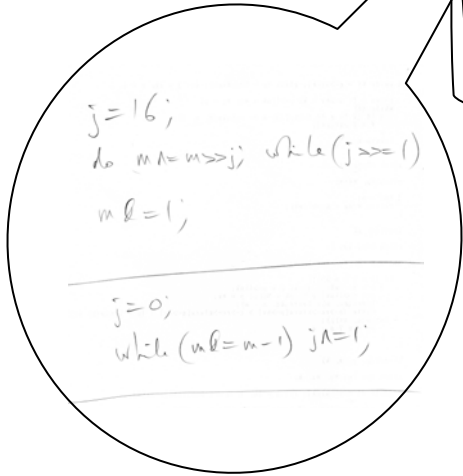


Helen Sharp

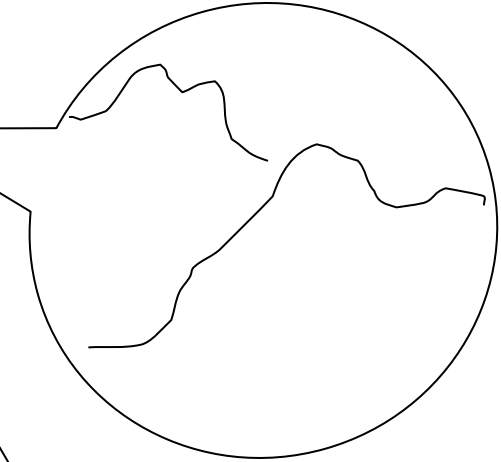




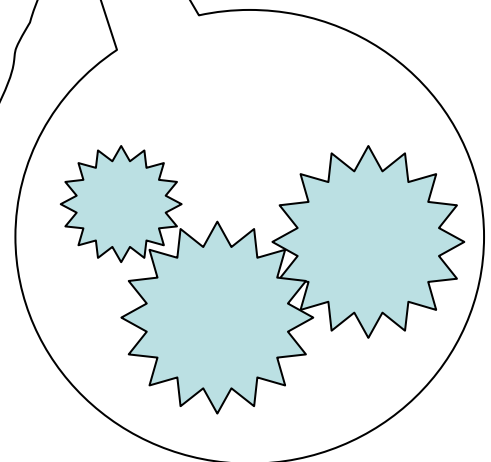
surfaces



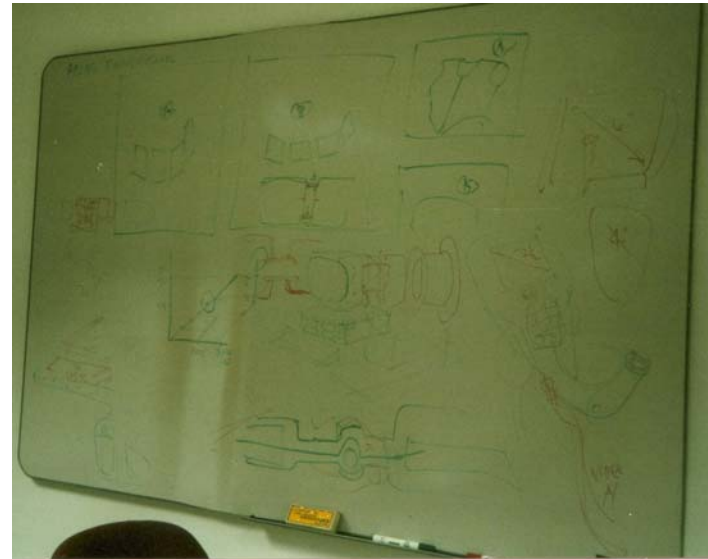
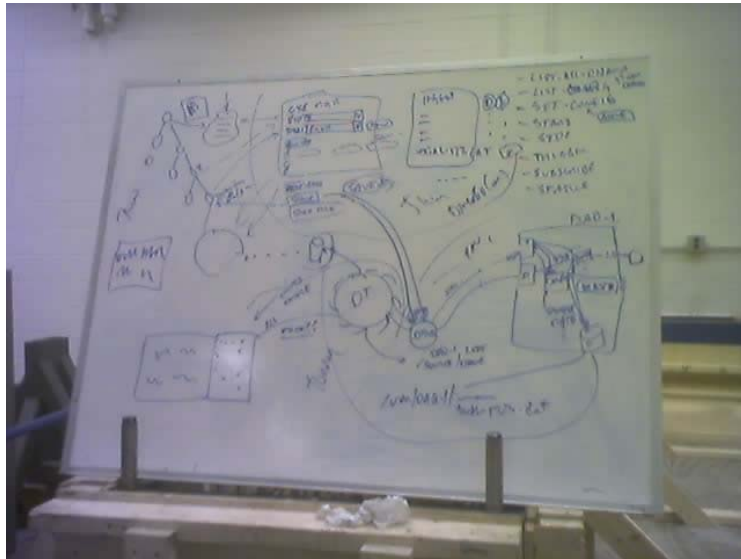
text with animation



landscapes



abstract machines



disciplines of innovation:

- ways to maintain the knowledge base
- ways to change perspective
- ways to expand the search space

Many of their strategies concern
expansion of the design space,
not just convergence to a solution.



The Open
University

Centre for
Research in Computing

simplification (solving a simpler problem)

transformation (into another, easier to handle, form)

re-segmentation (dividing it up differently)

relaxing constraints (changing reality)

analogy (using differences to explore boundaries)

abstraction (solving the essence)

re-shaping the problem space (solving the problem instead of the solution)

seeking insight (rather than solution *per se*)

nibbling (working at small bits that *can* be solved)

Experts use systematic practices ...

- testing, debugging, code reviews
- daily discussions
- building tools to suit practice
- disciplines of innovation
- play, tinkering, bricolage

... that are socially embedded and reinforced

- pair debugging
- reliance on team to catch slips
→ freedom to experiment
- rewarding success
- roles related to skills





"heroic programming"
the guy w/ the award
who developed s/w that
worked for as long as he
stayed at the company and
stopped the day after

doing certain actions in the right order:

e.g.:

“Setting goals before taking action

Understand problems before generating solutions

Designing before writing design documents

Validating designs before investing in code

Steak before sizzle”

John Schrag, Autodesk

<http://dux.typepad.com/dux/2009/07/values-in-software-design-practice-.html>

exploration

- creativity
- provisionality
- design alternatives
- the role of error
- tinkering / bricolage / play





TDI Blog | Design: Metserve Ent.



The Open
University

Centre for
Research in Computing

cognitive issues

- **abstraction** is hard, and comes at a cognitive cost
- **bias** in perception, attention, cognition, judgment
- the need for **operational models**

experts can:

- identify what is relevant and important and to ignore the unimportant;
- match strategies to tasks;
- recognise resonances across domains;
- have and use strategies for dealing with intractable problems by recognising analogies or transforming them into simpler problems;
- understand the consequences of design decisions, to encompass both abstraction and detail;
- handle conflicts among constraints or principles.



reflective practice

- systematic efforts to alter perspective
- deliberate changes of representations, of paradigms
- cultivating an awareness of alternatives
- reviews of experience
- tolerance for error

“Sound methodology can empower and liberate the creative mind; it cannot inflame or inspire the drudge.”

(Brooks)



The Open
University

Centre for
Research in Computing