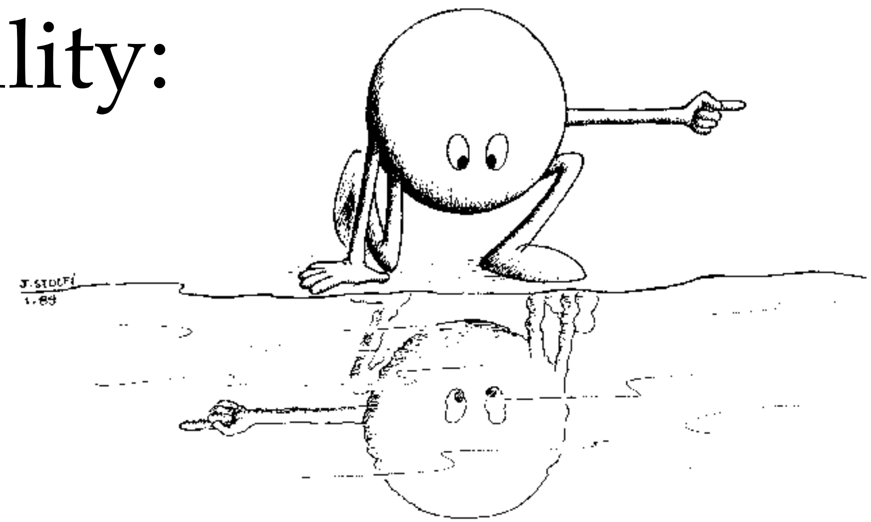
Sensor and sense-ability: Building systems in the face of uncertainty



"Where theory meets practice..."

Simon Dobson and Juan Ye
School of Computer Science, University of St Andrews UK

sd@cs.st-andrews.ac.uk
http://www.simondobson.org

ye@cs.st-andrews.ac.uk
http://sites.google.com/site/juanyeresearch/

Overview

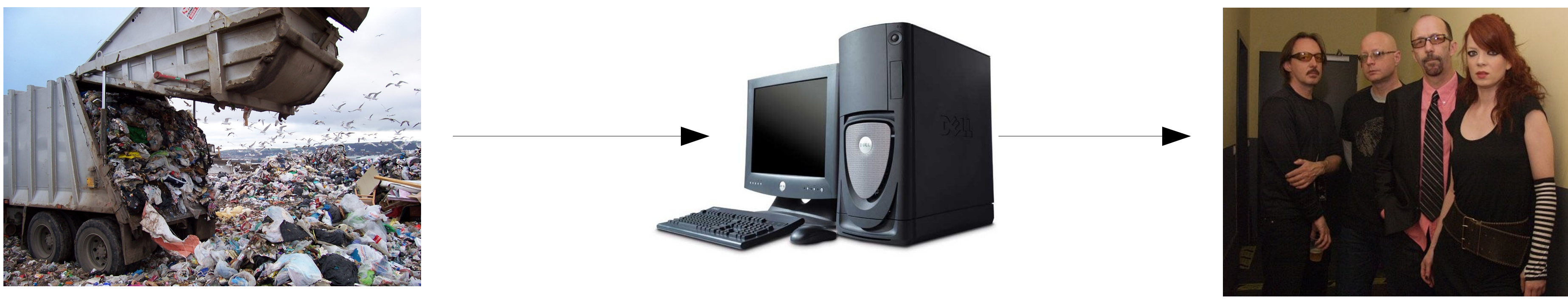
- An emerging class of systems are driven by data sensed directly from the real world
  - Adapt and/or exhibit behaviour without detailed human control
  - Uncertain and imprecise inputs, consistent output
- How should we program these systems?
- Our aim
  - Explore past work and future challenges in building sensorised, context-aware adaptive systems

The place of computer science

- The new microscope
  - The "third pillar" alongside theory and experiment
  - Simulation, sensors, visualisation, …
- Foundational understanding
  - Formal description of *how* a process operates
  - Languages, systems, network science, …
- Societal impact
  - Engineering complex systems reliably
  - Systems engineering, mobility, security, ...

GIGO

- "Garbage in, garbage out"
  - The wrong data will generate the wrong output



  - If the parameters don't meet the rely conditions, the results won't (always) meet the guarantee conditions

Not a new idea...

On two occasions I have been asked, "Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?" ... I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question.

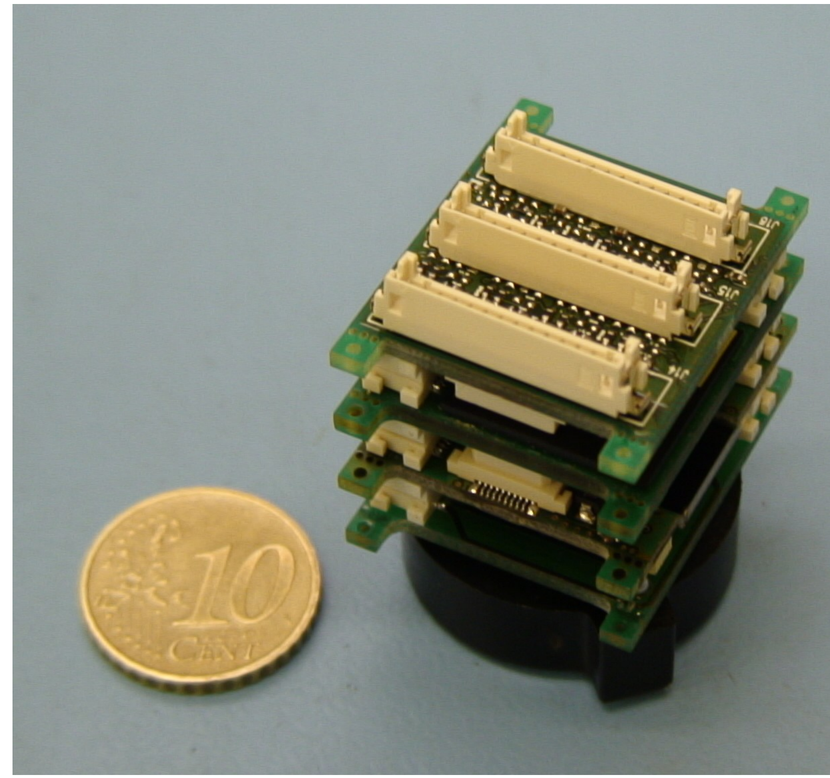
Charles Babbage. Passages from the Life of a Philosopher. 1864.

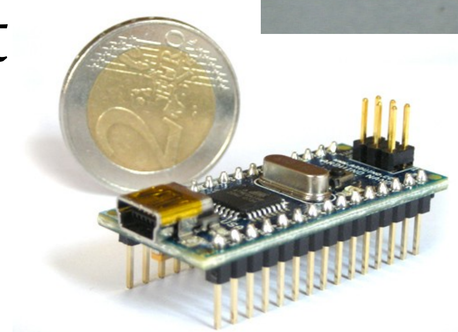
Really a confusion of ideas?

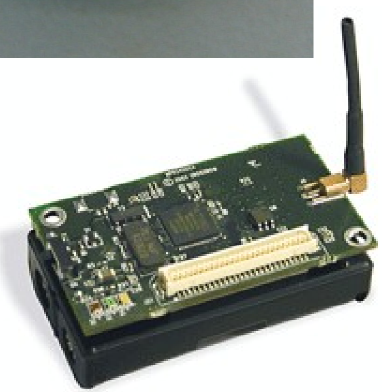
- Babbage's assertion perhaps reflects a scientific determinism we no longer share
  - Heisenberg uncertainty, chaotic dynamics, ...
- We are used to the idea that systems come with *inherent* uncertainty
  - Can't be engineered away
  - Systems must still behave "correctly" – even if their inputs are "garbage" (or at least imperfect)

Sensor networks – 1

- Sensor networks
  - Lots of small "motes"
  - Simple processing, communications, memory
  - Low-power
- Collect data from their environment and return to a base station

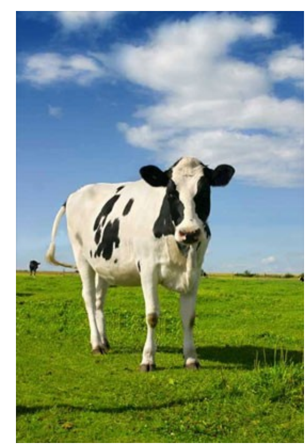






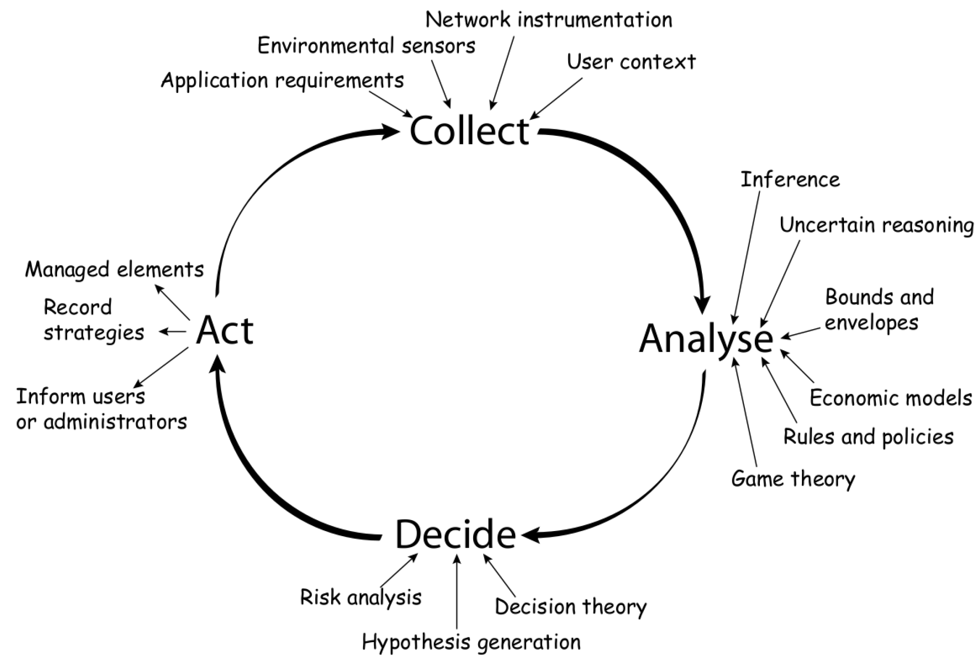
Sensor networks – 2

- Are – and will remain – challenging
  - Don't get a Moore's Law effect to improve performance over time
  - Sensors have limited precision, accuracy and temporal resolution
  - Node failure is unexceptional
  - Network must survive interruptions (although individual nodes won't)



Control

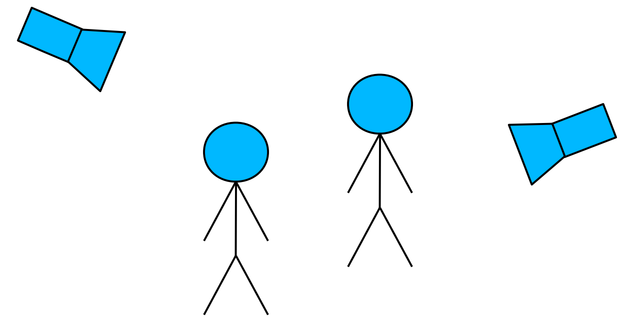
- Often need to do adaptive control in these environments
  - Change mode, duty cycle, processing, …
  - Ensure scientific (mission) goals are maintained across adaptations
- Basis for control is (imprecise) measurement



Dobson *et alia*. A survey of autonomic communications. ACM Trans. Auto. Adapt. Sys **1**(2). 2006.

Sensor-driven activity

- Increasing "sensorisation" of the environment
- Drive action directly from sensed values
  - Data is *evidence* of fact, not fact
  - Noise makes exact determination problematic
- Match observations against a *model* of what we *expect* to observe
  - Leverage the structure of behaviour



Sensors may see some, all or no people; agree or disagree on their identities; repeat observations; report with different footprints and frequencies

Noise is (often) random; phenomena of interest (often) aren't

The rest of this talk

To what extent *can* we continue to generate the right answers from the wrong figures?

- Programming in the presence of uncertainty
  - Represent data in a form suitable for open-ended reasoning tasks
  - Resolve inconsistencies, tolerate small (and essentially unavoidable) errors in sensing etc
  - What are the appropriate programming structures for this environment?

Systems thinking

ACT

Increase sampling rate; communicate with nearby animals to see of they're doing the same thing

SENSE

What can we sense? Can we get evidence from different sources?

What do we know about the animal's behaviour? How does the sensor data back-up hypotheses about what's happening? What else *might* cause this observation?

ANALYSE

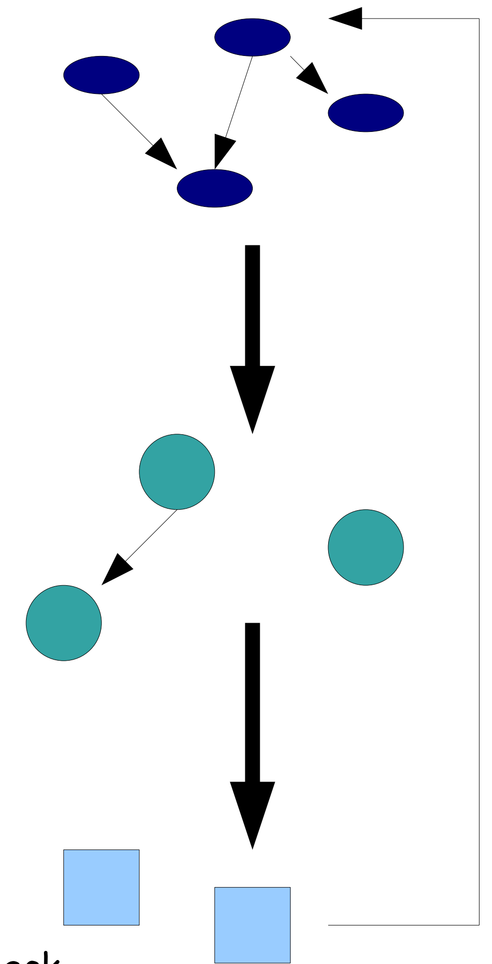
State of the seal

The seal is hunting for food, and diving really quickly

DECIDE

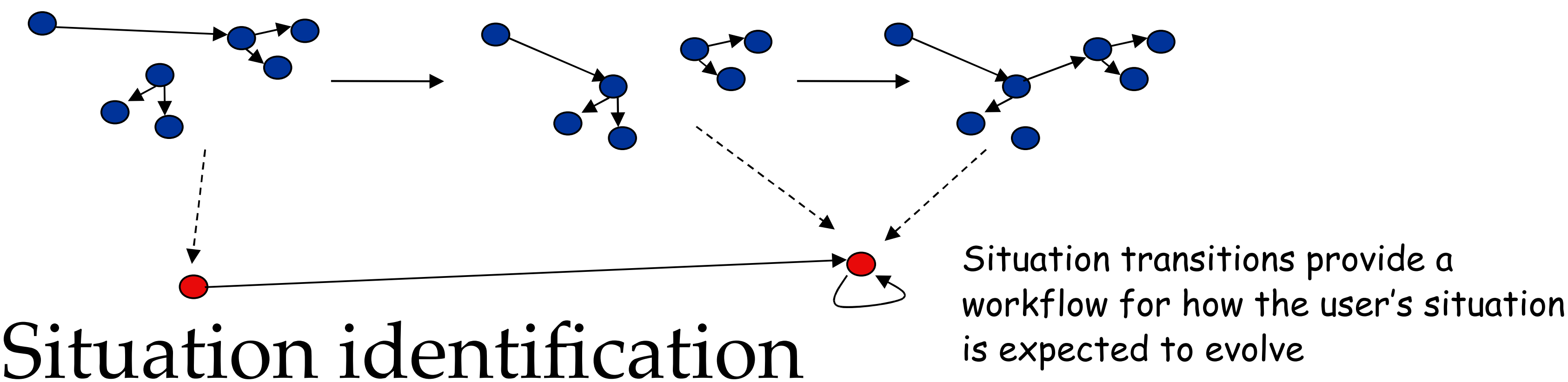
Context and situations – 1

- In pervasive computing there are a wide variety of definitions for the core concepts
  - *Context*: the environment in which a system operates, understood symbolically
  - *Situation*: an interpretation of the current context in terms of an expectation model of the world
  - *Behaviour*: the observables arising from the system's responses



Context and situations – 2

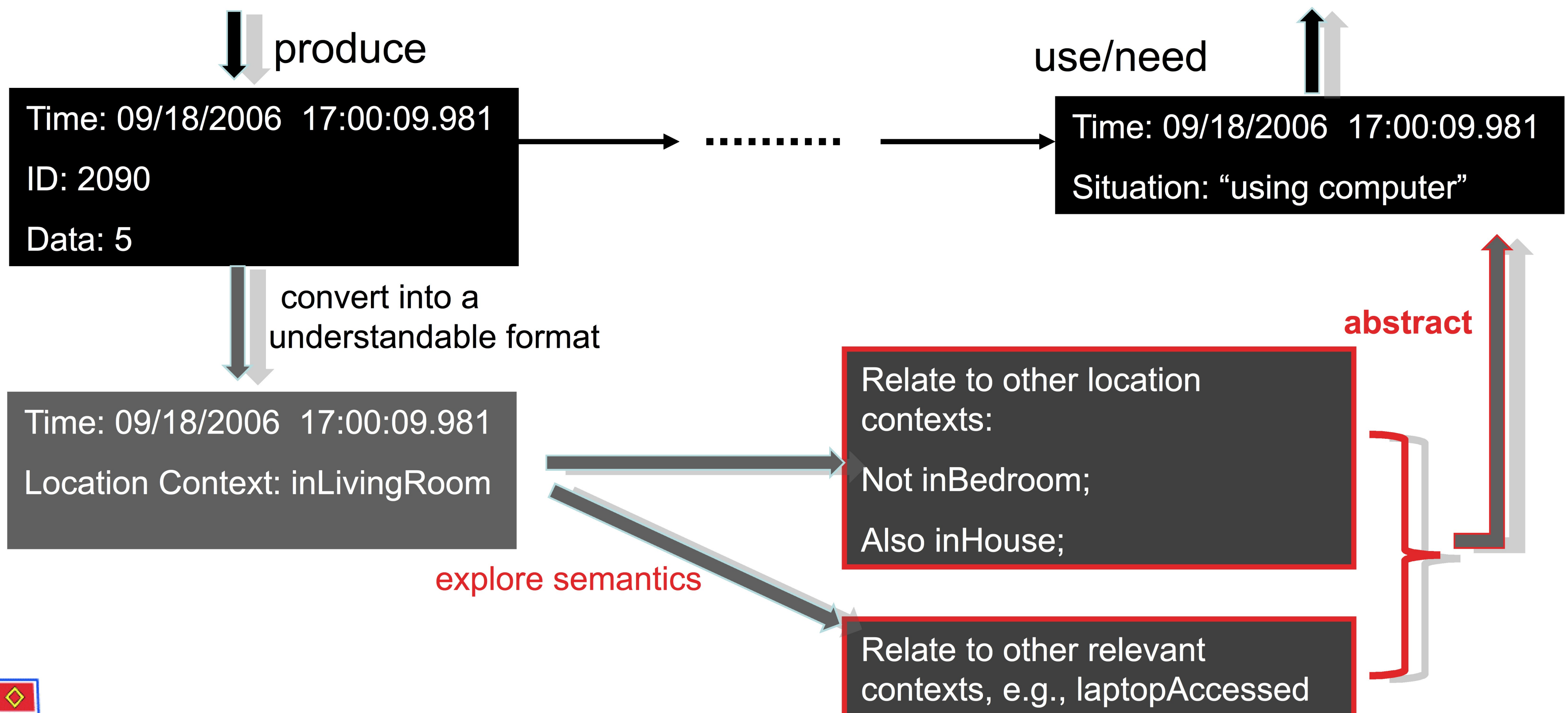
- Context is often redundant and conflicting
  - Many different contexts determine the *same* information (situation)



- Situation identification
  - Semantic: given a context, what situation are we in?
  - Programming: how do we make this decision?

Why not work with context?

- Situations are closer to how designers think about systems



Example: location

- Surprisingly (or perhaps not) subtle domain

Co-ordinates and named spaces

- "At 55deg3minN, 3deg45minW"
- "In A1.15"

Unknown

- "No idea"

By negation

- "Not …"

Non-located task

- "Out/on holiday"

Proxy

- "His badge was last seen at …"

Located task

- "Meeting Willard"

Functional spaces

- "In a conference room"
- "In his office"
- "In Willard's office"
- "In his car"

Temporal

- "At 1000 he will be…"
- "At 0800 he was…"

Default

- "At this time he is often/usually at …"

Relative

- "With Willard"

Spatial

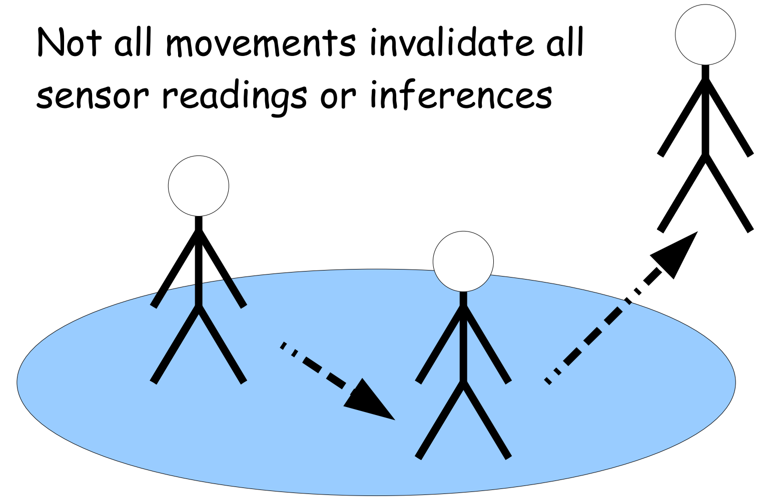
- "Within 250m of…"
- "Between … and …"
- Either at … or … or …"

Dobson. Leveraging the subtleties of location. Proc. sOc-EUSAI'05. 2005.

Sources of uncertainty

- Dynamism
  - People move
- Engineering
  - Precision, accuracy, timeliness, calibration
- Inference
  - Track the imprecision
  - Recognise uncertainty in conclusions explicitly

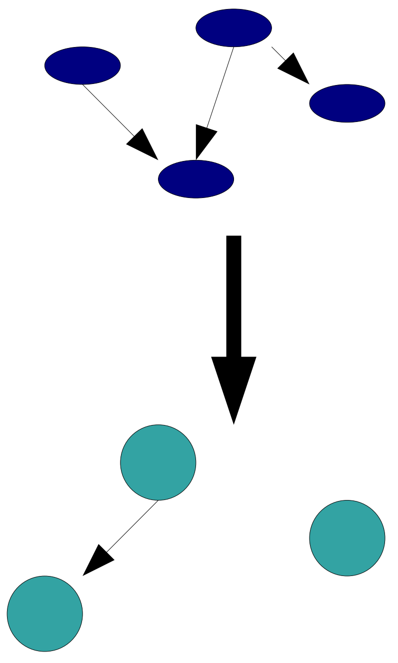
Not all movements invalidate all sensor readings or inferences



```
example:reading
  a sensor:Observation ;
  example:about ubitag:010131789 ;
  sensor:observedAt [...] ;
  sensor:temporalDimension [...] ;
  sensor:observedBy example:CASLUbisense ;
  sensor:value
    [ a location:Coordinate ;
      location:referenceCoordinateSystem
        example:ubisenseCoordinateSystem ;
        location:x "1.15" ;
        location:y "3.67" ;
        location:z "21.35"
    ] .
```

Stevenson *et alia*. ONTONYM: a collection of upper ontologies for pervasive application development. Proc. CIAO'09.

Approaches



- Predicates
  - What ranges of data map to what
- Bayesian inference
  - P(S | C) – being in situation given a particular set of observations
- Dempster-Schafer evidence theory
  - Distribute mass of belief
- Case-based reasoning
  - Use similarity to past, human-classified cases

Sources of knowledge

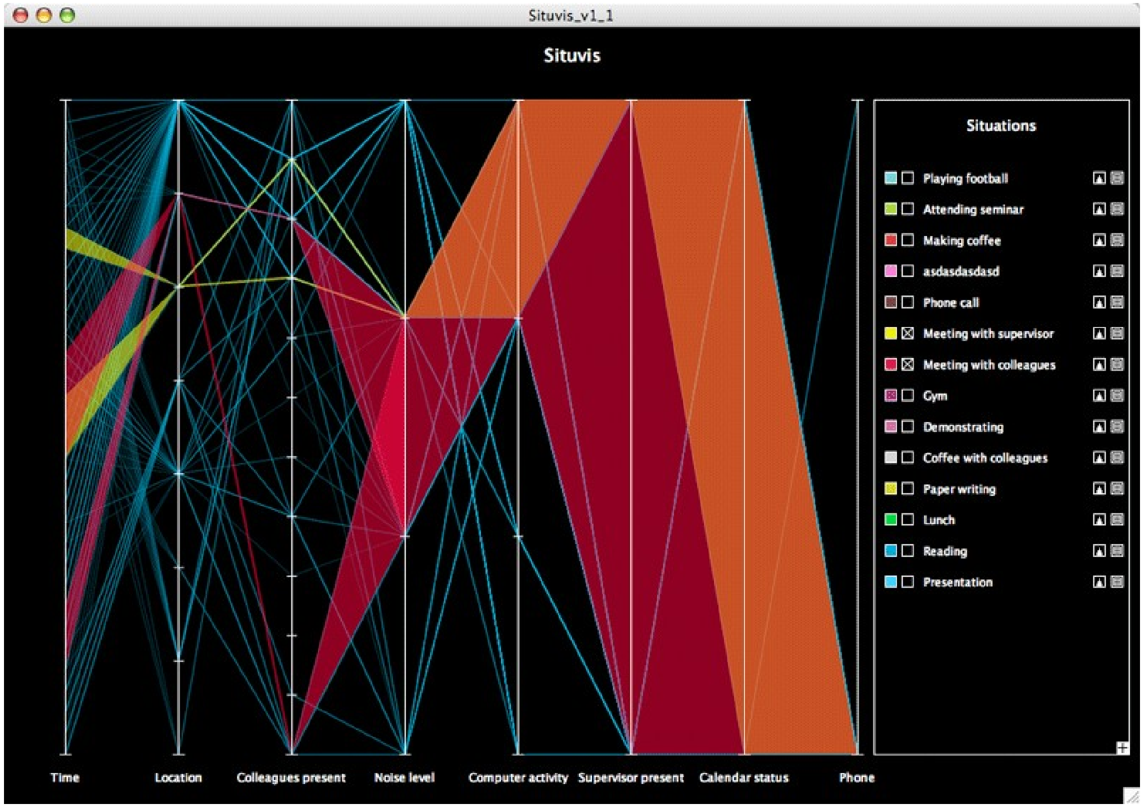
- Human understanding
  - Possible, impossible, liklihood
- Data sets
  - Future will be like the past (?)
  - Learn patterns from past observation
  - Precision, recall, F-measure

Only classify rates broadly

There is a critical shortage of good, clean, marked-up data sets

Situvis

- Exploratory specification of predicates
  - Visualise how system would respond

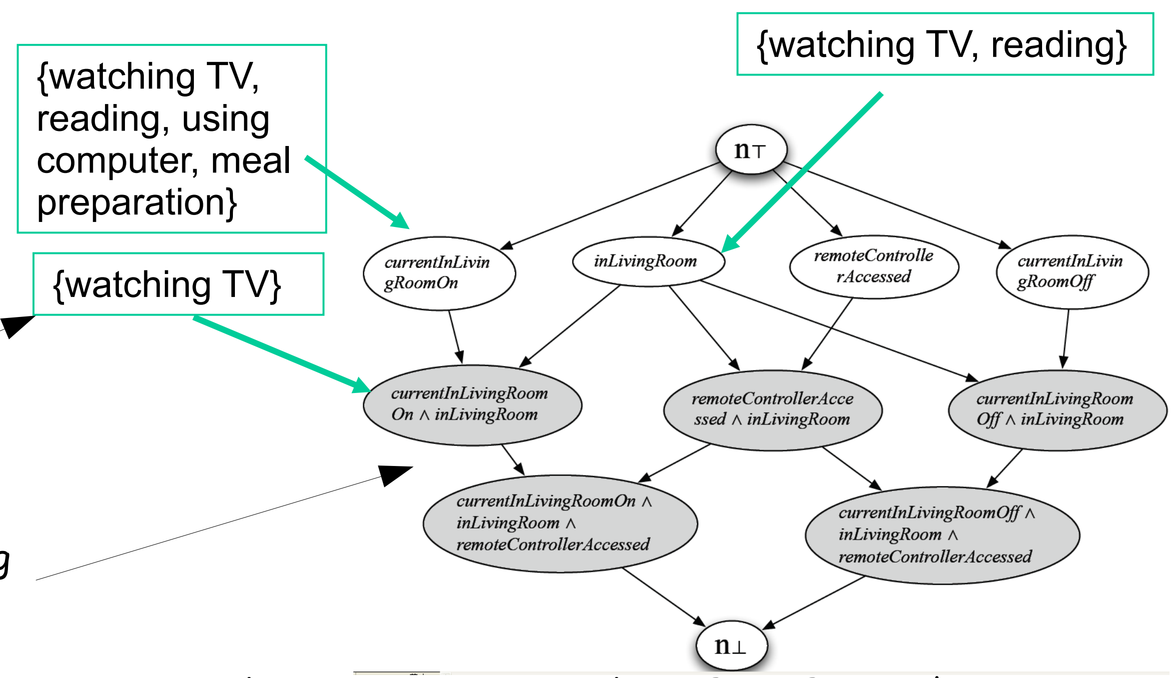


Clear *et alia*. Situvis: a sensor data analysis and abstraction tool for pervasive computing systems. Pervasive and Mobile Computing. 2010.

http://www.situvis.com

Structuring situations

- Situations have structure
  - "Meeting" vs "meeting with Erica" vs "Group meeting" vs …
  - Capture this using a lattice relating observations to the situations they are consistent with



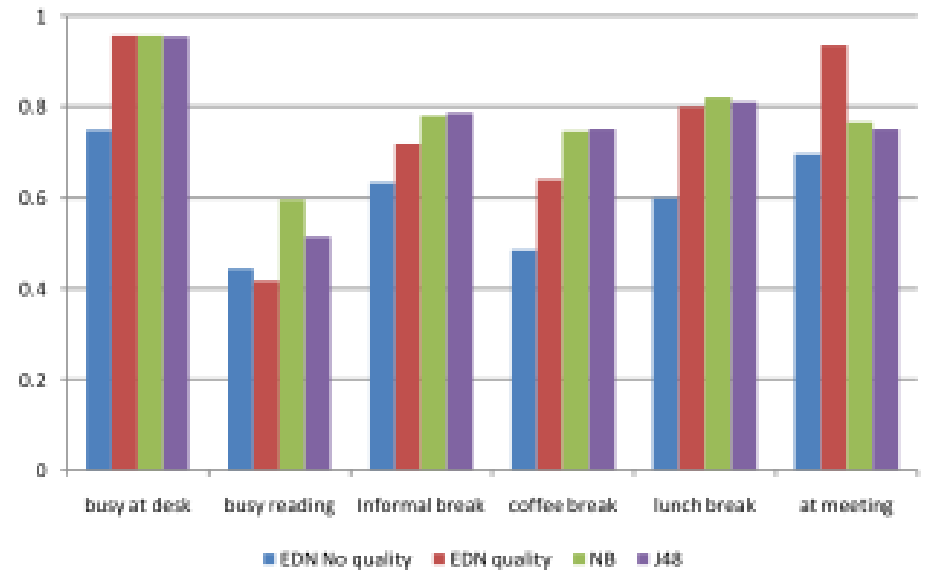
Consistent with a *range* of possible situations

Collections of observed events provide supporting evidence

Ye *et alia*. Using situation lattices in sensor analysis. Proc. Percom'09.

Profile of results

- All these methods tend to identify particular classes of situations well – but not all



- Is there a “best” method?

Impact

- How *unlike* normal programming!
  - Not sure what condition we're in
  - ...therefore can't decide certainly on what behaviour we should exhibit
  - Data comes with provenance
  - ...and with unusual types, with non-subsumptive relationships
- How should we best present this new domain to software developers?

Programming challenges

- Stability
  - Errors must damp-down inherently
- Multiple possibilities
  - Accept multiple behaviours, and their overlaps
- Reversing
  - All decisions are tentative and must be undone or mitigated

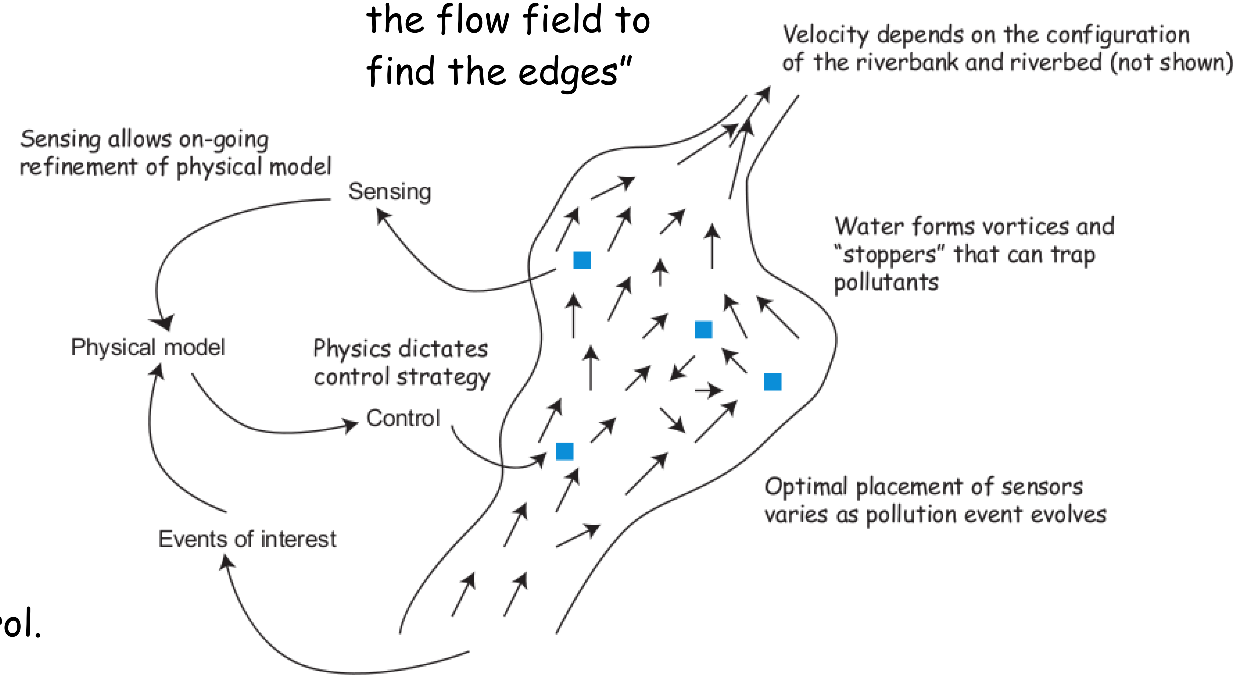
Mission languages

- Goal: capture the mission of an adaptive system
  - The *raison d'être* for which it is deployed
  - The parameters it's allowed to adapt, and limits
  - The tactics it can deploy

"Swim outward against the curl of the flow field to find the edges"

"Maximise the lifetime value of each node"

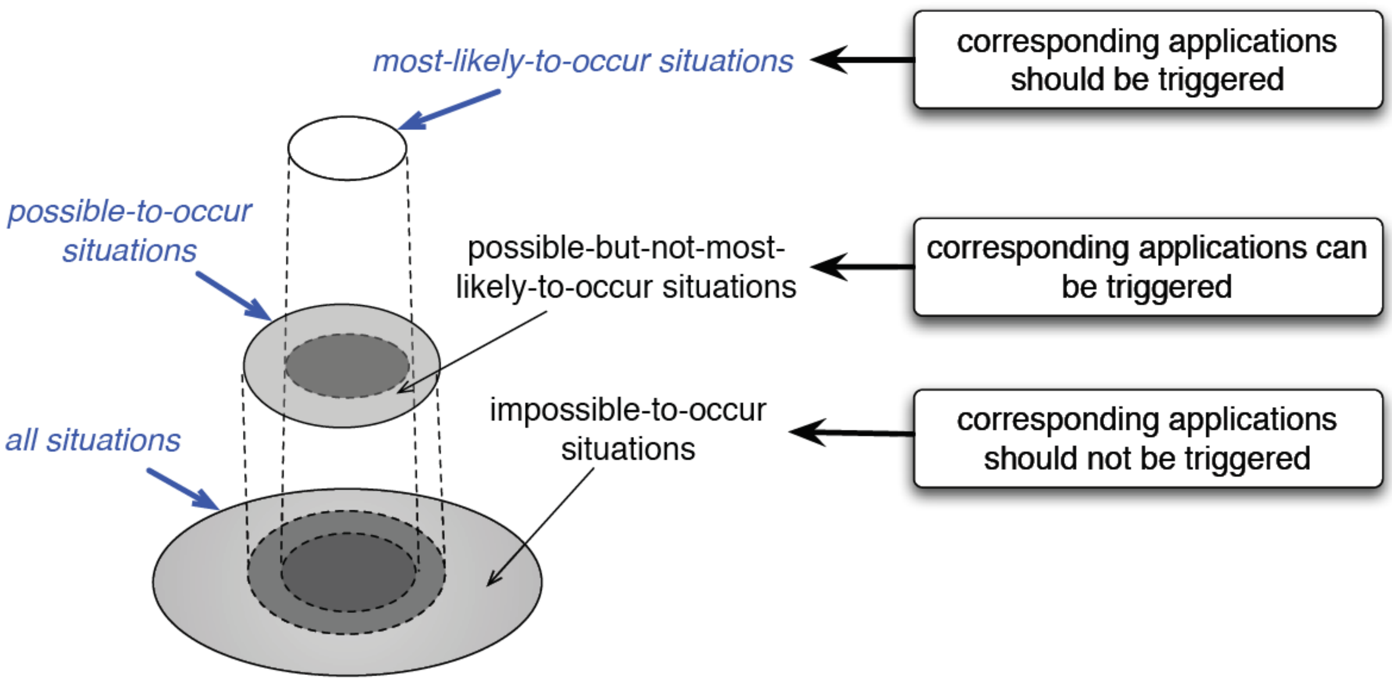
"(Re-)deploy appropriate resources to each event"



Dobson *et alia*. From physical models to well-founded control. Proc. IEEE EASe. 2009.

Behaviour

- Can't usually narrow-down to exactly what's being observed
  - Impossible, possible, most likely



May be able to divide-up behaviour more finely, *e.g.* active intersection of behaviour

No ifs

- Decisions are less crisp
  - How certain is “certain enough”?
- Thresholding throws away the weight of the evidence
- Weight may change rapidly
  - Make a decision, plan how to reverse it later
  - Truth- or confidence-maintenance

5 things to take away

- Can't avoid encountering uncertain data with complex provenance
- Embrace it: it's better than assuming things are different than they are
- Can capture a lot of uncertainty generically
- Programming involves identifying possible and consistent situations
- Needs new constructs and languages that match the domains of modern interest

Acknowledgements

- None of these ideas are truly mine, and all have benefitted from the insights of colleagues
  - Dr Juan Ye, Dr Adrian Clear, Dr Stephen Knox, Dr MA Razzaque, Dr Hui Zhang, Dr Emerson Loureiro, Dr Michael Collins, Dr Steve Neely, Graeme Stevenson, Lei Fang
  - Prof Paddy Nixon, Prof Aaron Quigley, Prof Al Dearle, Prof Franco Zambonelli, Prof Giovanna Di Marzo, Prof Alois Ferscha, Prof John Strassner, Dr Mirko Viroli, Dr Marco Mamei, Dr Joel Fleck, Roy Sterritt