University of St Andrews | FOUNDED 1413

Making sense of sensing

DPhil Computer Science,
University of York, 1993

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

simon.dobson@st-andrews.ac.uk
http://www.simondobson.org

Overview

- We're seeing more and more sensor systems
  - *Report* accurate values for some observed phenomenon (temperature, humidity, …)
  - *Classify* observations into human-recognisable categories (cooking, cleaning, burgling, …)
  - Direct input to algorithms, no human in the loop
- My goal
  - The characteristics of sensor-driven systems
  - Addressing some specific challenges

Multi-platform

- Lots of computational elements
  - Sensor nodes
  - Sinks and backhaul
  - In the cloud

- Now hearing a lot about *fog computing*
  - Locate processing at the "right" level
  - Not clear where is "right"
  - Programming approach is completely different at each level: hard to migrate functionality
  - And anyway this sort of misses the point...

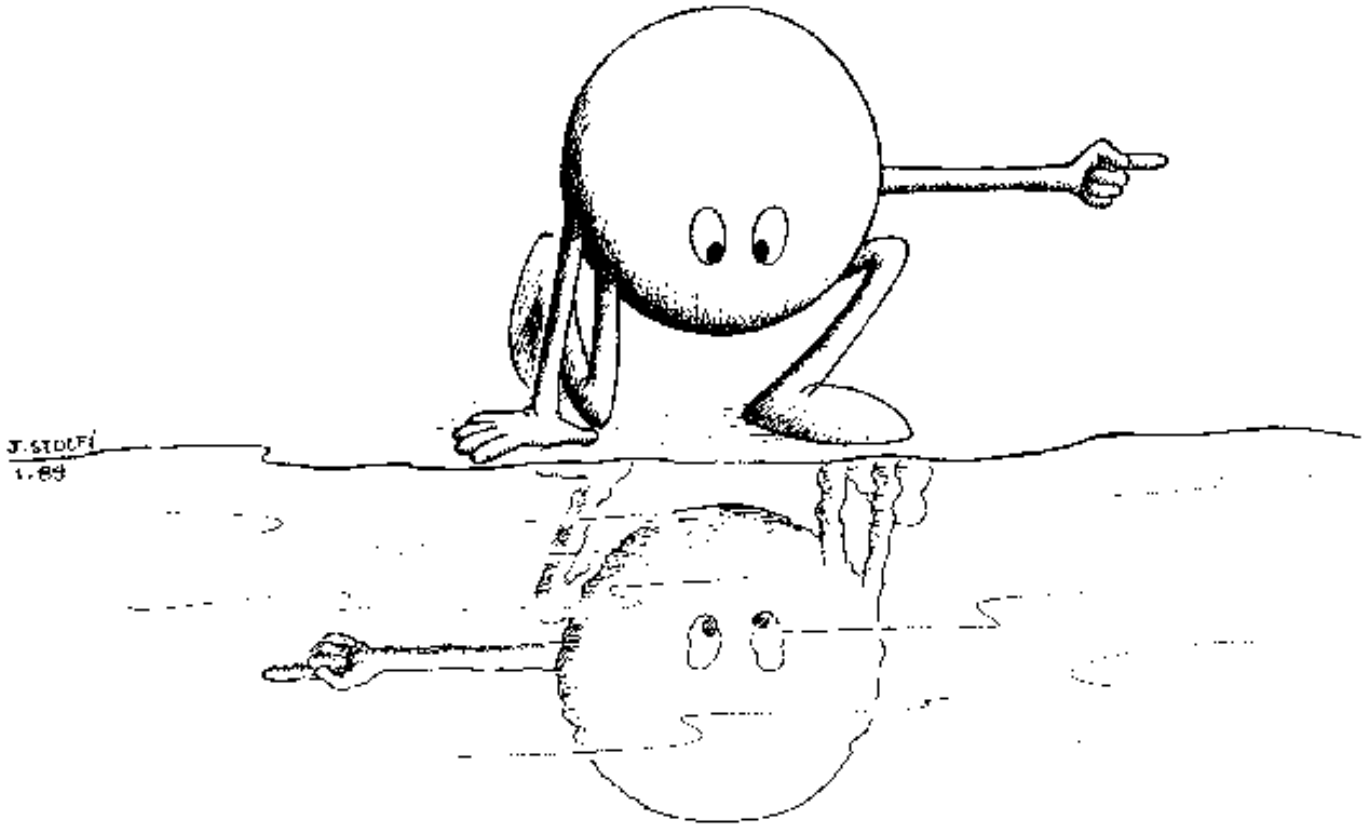
The sensor challenge

- Treating sensor data as input like any other
  - Model with techniques like process algebras, DTMCs, …
- But sensor data *just isn't like that*
  - Environmental challenges and exposed equipment
  - Leads to a collection of unusual failure modes
  - Responding to the input means also responding to the junk data that's interleaved with it

The authors of one famous early experiment (Great Duck Island, 2002) deemed 30—60% of their sensor data to be junk

Image from lighthousefriends.,com

When theory meets practice

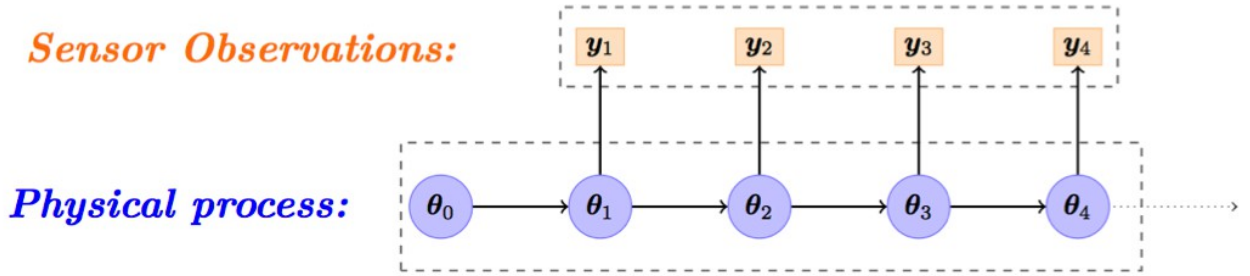


In theory, there is no difference between theory and practice. But, in practice, there is.

Jan L.A. van de Snepscheut

A stochastic approach

- Re-conceptualise sensors as evidence-providers rather than data- or value-providers
  - Use to confirm/refute model hypotheses
  - Learn the distributions being observed
- One way is to view the system as a Hidden Markov process
  - Works with multiple sensors correlated with each other watching the same phenomenon



Some of the main challenges

- Noise
  - Difficult/impossible to engineer away
  - Deal with *confidence* of the *most probable* signal
- Confounding variables
  - Things happening that we don't know about
  - Can we separate the different causes?
- Not knowing what to look for
  - Classifying signals/events into activities/situations
  - What are the situations we're interested in?

There's nothing in any way canonical about these, they're just problems we happen to have been interested in over the last three years or so

Science of Sensor Systems Software

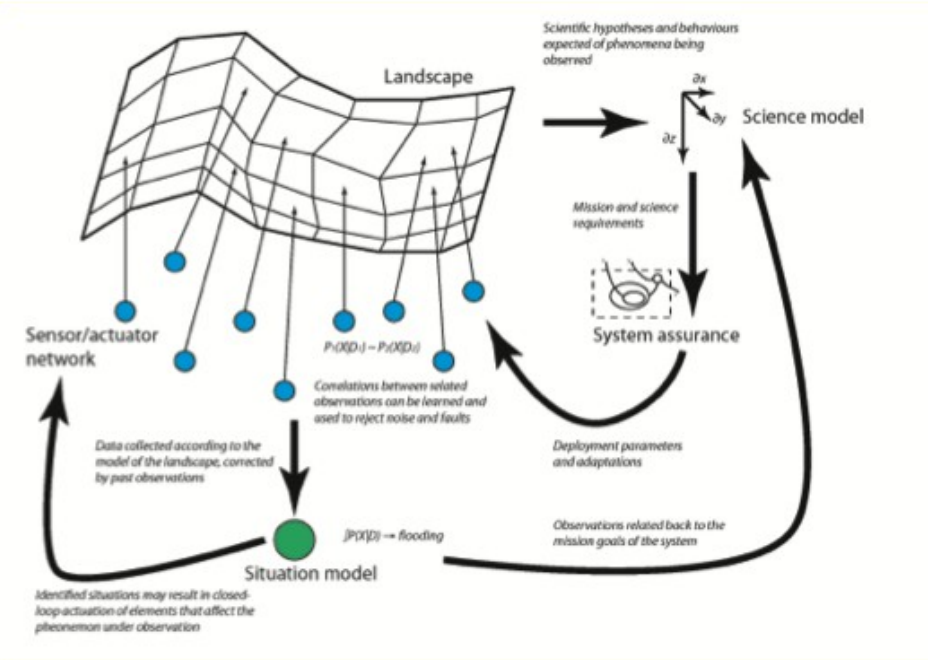
- A five-year, £5.2M EPSRC Programme Grant
  - "Vertical slice" from formal models, through verification and analysis, to deployment
  - Four universities, 10 commercial and agency partners

University of Glasgow

Imperial College London

University of St Andrews

UNIVERSITY OF LIVERPOOL



Noise

- An inherent uncertainty that can't be engineered out of a system
  - Physical degradation
  - Occlusion and fouling
  - Positional uncertainty
  - Interference, accidental or deliberate
  - ...and also describes lots of other data sources

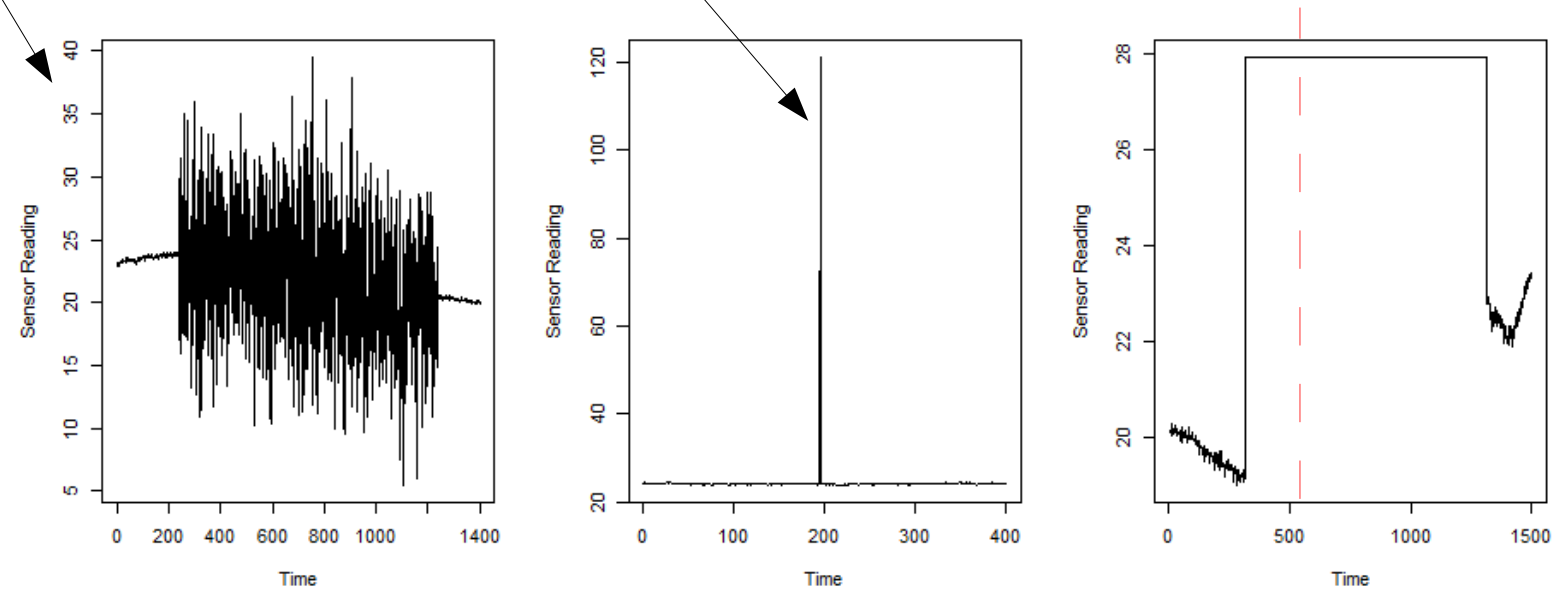
- *Physical* issues that give rise to *faults* in the data
  - Change over time, need autonomic management

Fault types

Natural variation plus noise

Not likely in your data?

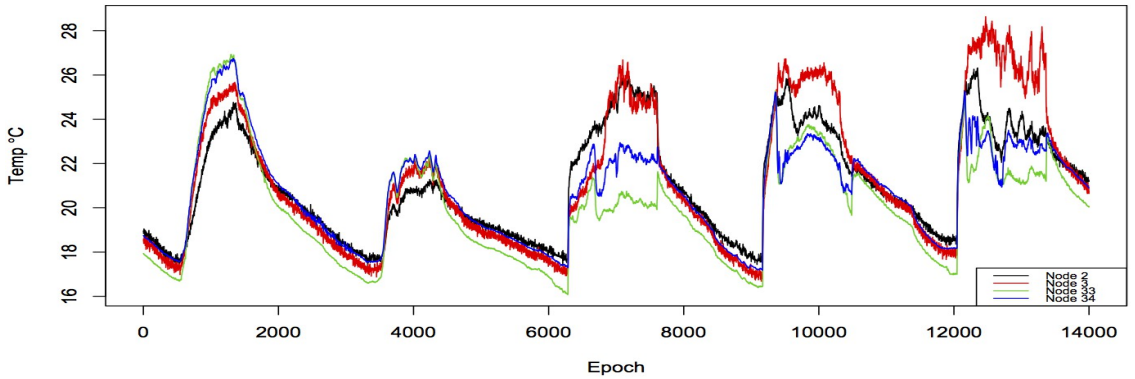
Can we tell here that this is an extended fault? Or is it a change of phenomenon?

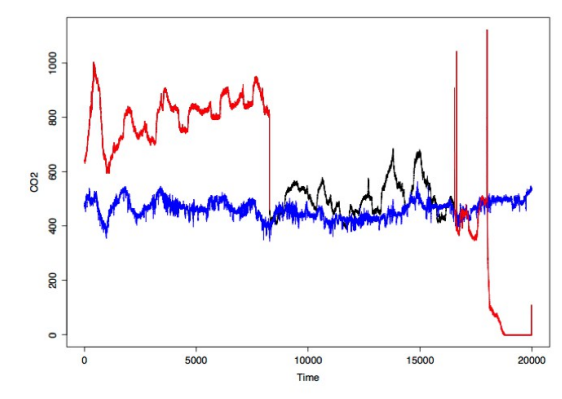


- Noise in the environment and the electronics
- Point (or wider) spikes
- De-calibration (drift) in space and time

Multi-sensor systems

- Neighbouring nodes observing "the same" phenomenon





- Look at the differences between them to learn the ways in which the true signal is being convolved with noise

Bayesian Sequential Learning

- Learning = sequential model update

$$
\begin{aligned}
p(\theta|\mathcal{D}_n) &= p(\theta|\mathcal{D}_{n-1}, e_n) \\
&\propto p(e_n|\mathcal{D}_{n-1}, \theta) p(\theta|\mathcal{D}_{n-1}) \\
&= \underbrace{p(e_n|\theta)}_{\text{likelihood}} \underbrace{p(\theta|\mathcal{D}_{n-1})}_{\text{prior}}
\end{aligned}
$$

The model given what's been observed up to (and including) now

The error, given the current observation

The observation, given what's gone (strictly) before

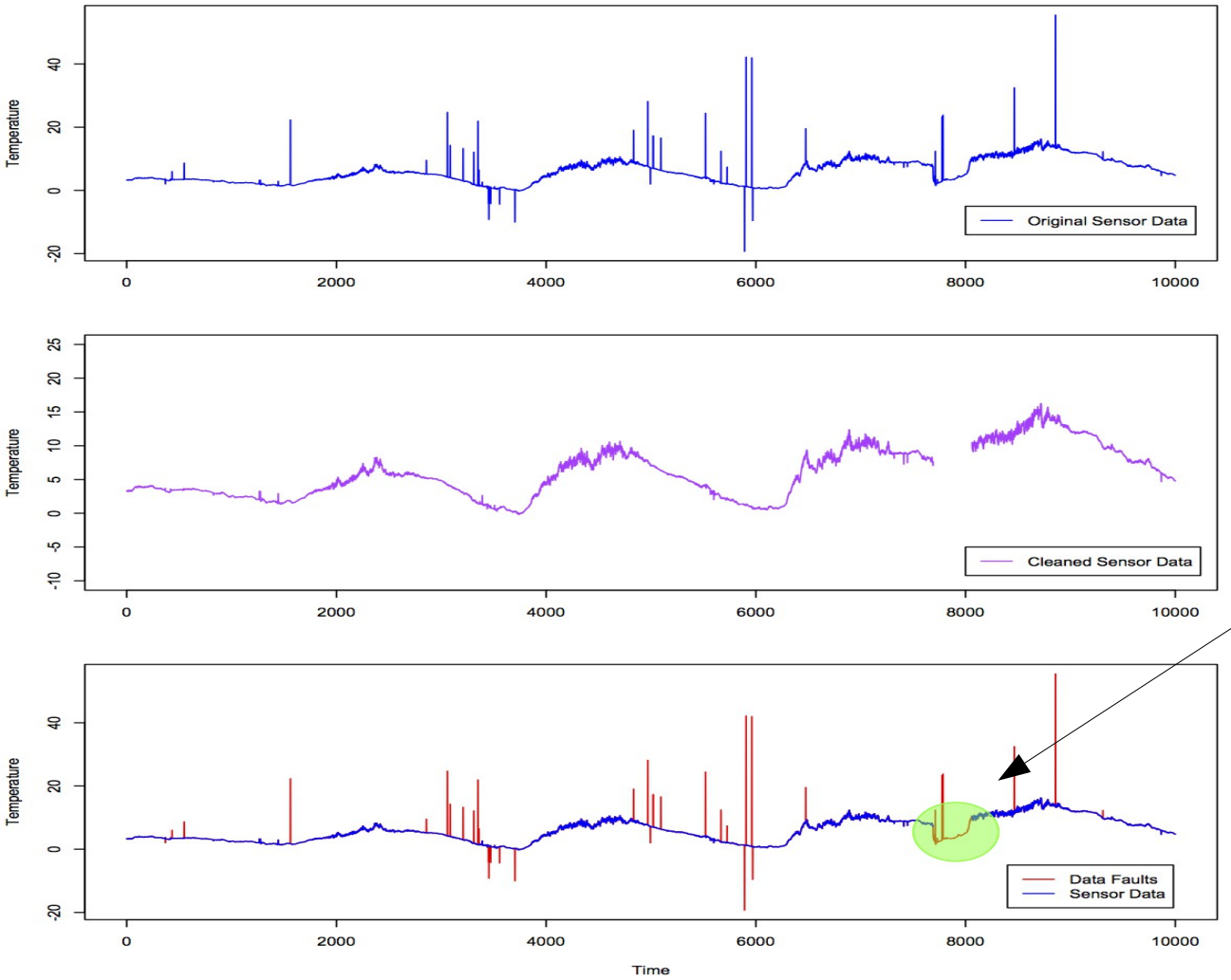
- Advantages
  - Efficient, constant space
  - Robust: test data against predicted distribution

Fang and Dobson. Data collection with in-network fault detection based on spatial correlation. Proc. ICCAC. 2014.

Example – 1

Original data

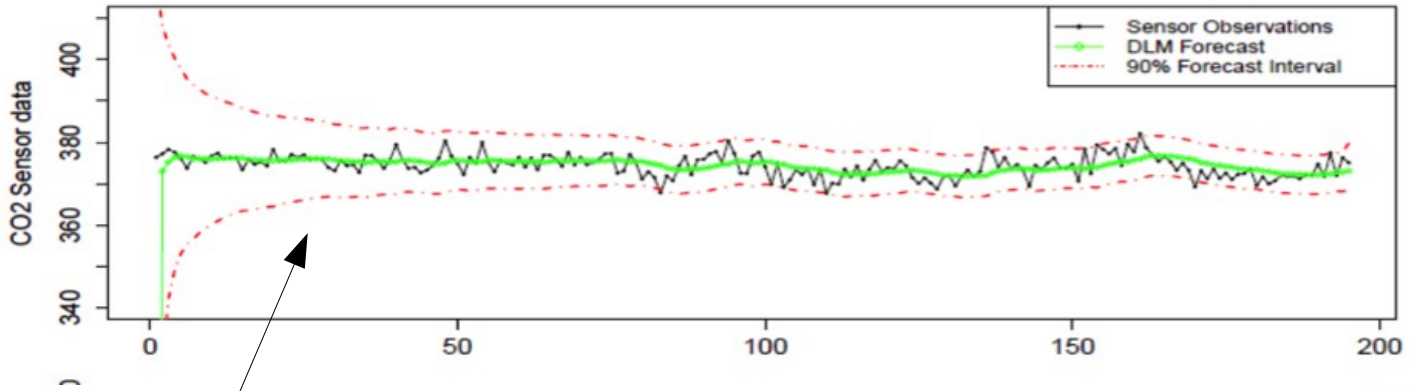
Cleaned data



May be a true observation, but not verified by neighbours. Can't tell without ground truth – which of course we don't have

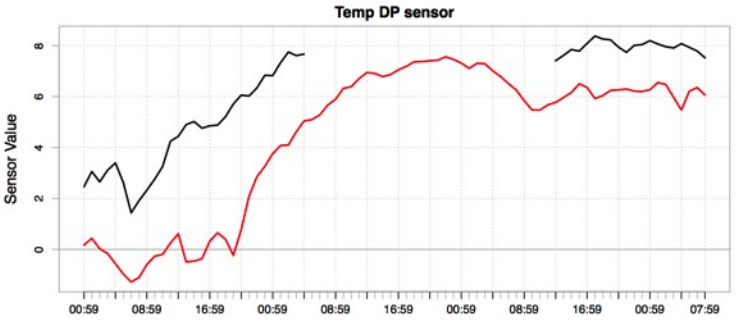
Data from the Lausanne Urban Canopy Experiment (2006)

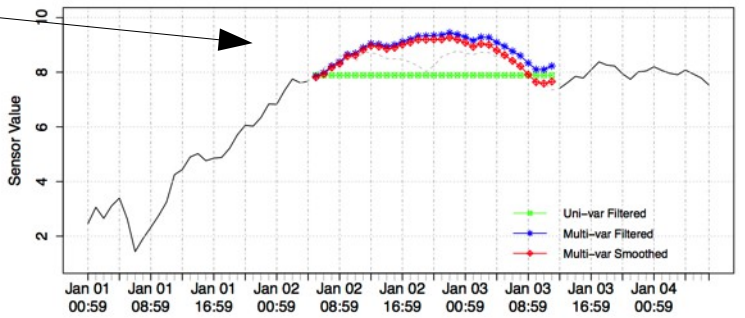
Example – 2



Use confidence intervals (rather than raw or smoothed signal) to trigger alarms

Re-generate "most probable" trajectory for missing signals, as correlated with data we *did* get





Data from Grangemouth facility (2016)

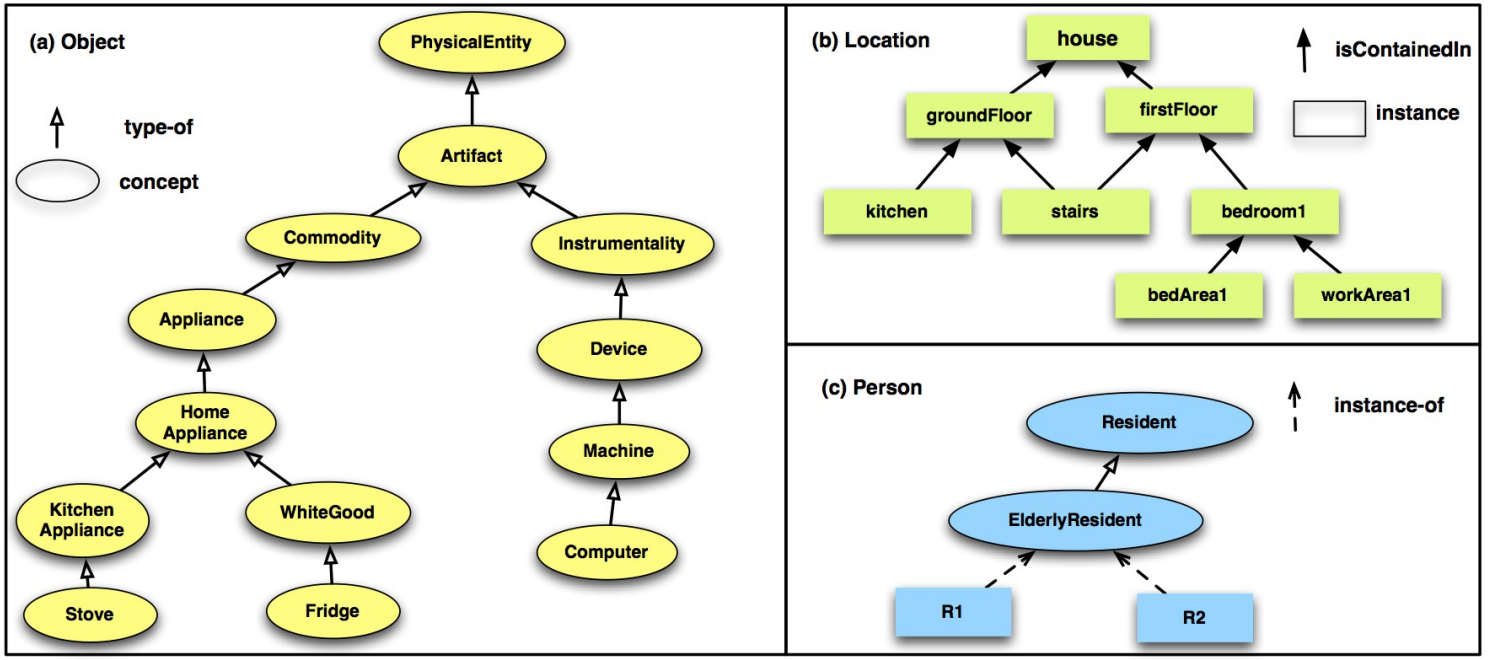
TOPOLYTICS™

Situation/activity recognition

- Many systems *classify* sensor traces
  - Given a sequence of observations, what activity is being observed?
    Ye, Dobson, and McKeever. Situation identification techniques in pervasive computing: a review. Per. Mob. Comp. **8**. 2012.
  - Very well-studied in the single-user case
- Often expressed as a stream of sensor events
  - Instantaneous observations
  - Fridge open, drawer open, entered kitchen, …
  - Typically not labelled with the actor
    "The door opened", not "Simon entered the room"
  - Given a stream of events, what's happening?
  - *Different* interleavings denoting the *same* activity

Ontological descriptions of events

- Descriptions of different system aspects
  - Describes the structure of a domain
  - Generalise place/object/person involved



Multiple targets

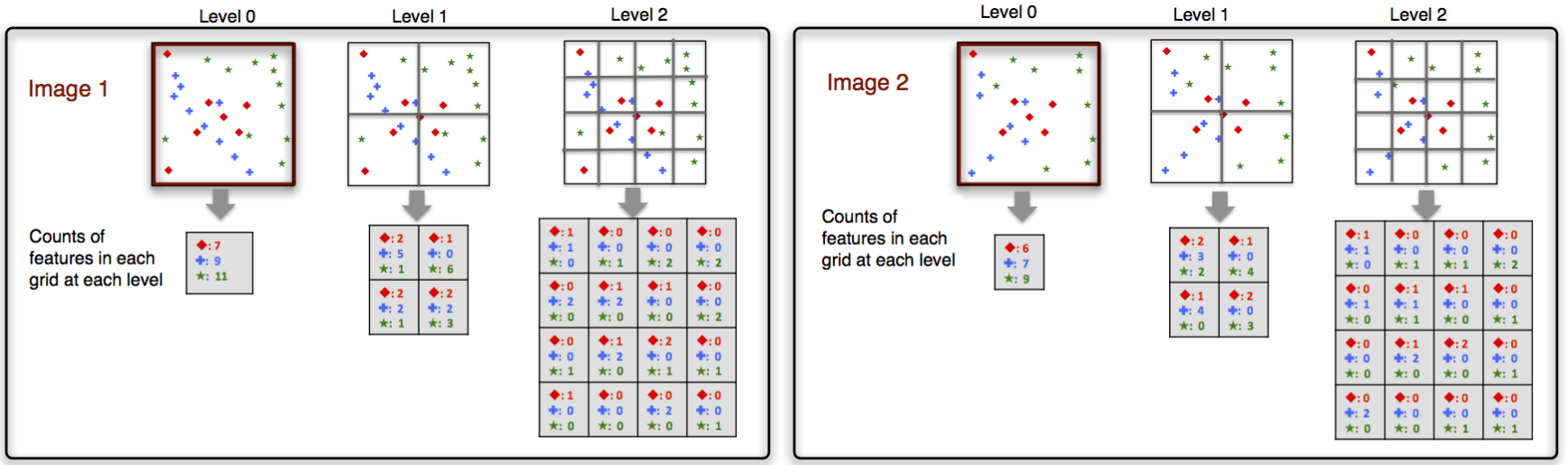
- But what if two (or more) activities are happening concurrently?
  - Making tea when the phone rings
  - One person making tea, another washing up
- Two (or more) activities giving rise to interleaved events
  - Extract fragments of event sub-sequences relating to the different simultaneous activities

Semantic distance

- Hardest case
  - Two people trying to do *the same* thing
  - No way to tell the events apart
- Easier case
  - Two people doing *different* things
  - Might expect (at least some of) the component events to be *different* in each case

Pyramid matching

- A way to define distances between images with sets of features



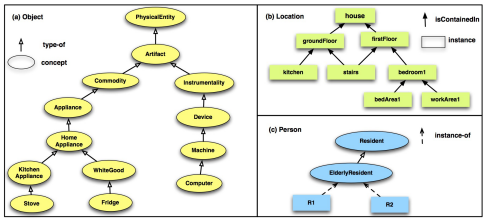
Match degrees between image 1 and 2 at different levels

Level 0	Level 1	Level 2	Penalised degrees
◆: 6 ✚: 7 ★: 9 Total: 22	◆: 6 ✚: 5 ★: 8 Total: 21	◆: 6 ✚: 5 ★: 7 Total: 18	18 + (21 – 18) /2 + (22 – 21) /4 = 19.75

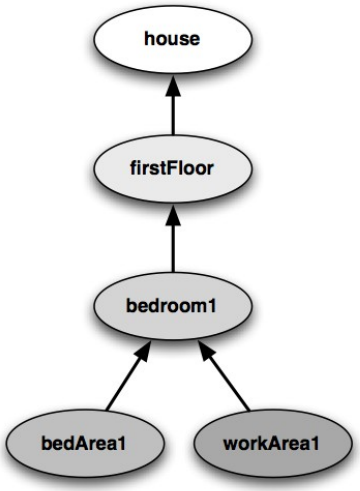
Grauman and Darrell. The Pyramid Match Kernel: Efficient learning with sets of features. J. Mach. Learn. Res. **8**. 2007.

KCAR

- Knowledge-driven Concurrent Activity Recognition



(a) *Hierarchy of domain concepts*



(b) *Calculate match degrees between sensor sequence and activities at each level*

Level 0 (2)	Sleep	6
	Work	6
	Wander	6
Level 1 (10)	Sleep	6
	Work	6
	Wander	6
Level 2 (20)	Sleep	6
	Work	6
	Wander	6
Level 3 (0)	Sleep	1
	Work	5
	Wander	4.8

(c) *Penalise match degrees at adjacent levels*

Level 1 to 0	Sleep	(6 - 6) / (20*10*2) = 0
	Work	(6 - 6) / (20*10*2) = 0
	Wander	(6 - 6) / (20*10*2) = 0
Level 2 to 1	Sleep	(6 - 6) / (20*10) = 0
	Work	(6 - 6) / (20*10) = 0
	Wander	(6 - 6) / (20*10) = 0
Level 3 to 2	Sleep	(6 - 1) / 20 = 0.25
	Work	(6 - 5) / 20 = 0.05
	Wander	(6 - 4.8) / 20 = 0.06
Level 3	Sleep	1
	Work	5
	Wander	4.8

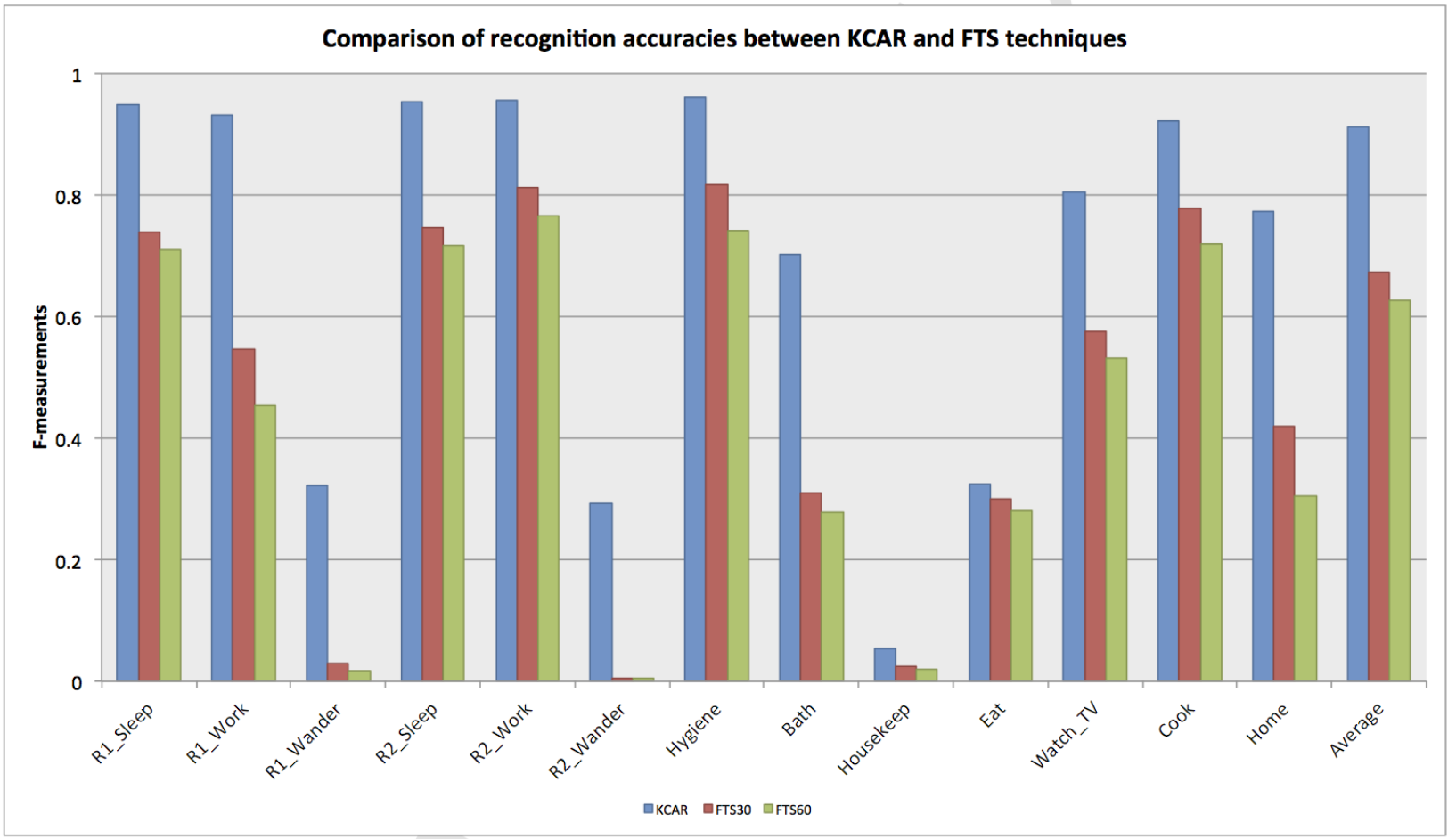
(d) *Sum penalised match degrees*

Sum	R1Sleeping	1 + 0.25 + 0 + 0 = 1.25
	R1Working	5 + 0.05 + 0 + 0 = 5.05
	R1Wandering	4.8 + 0.06 + 0 + 0 = 4.86

SAPERE
Self-Aware Pervasive Service Ecosystems
EU STREP Project
FET Proactive Initiave
FP7-ICT-2009.8.5: Self-awareness in Autonomic Systems

Ye, Stevenson, and Dobson. KCAR: A knowledge-driven approach for concurrent activity recognition. Per. Mob. Comp. **19**. 2015.

Results



Datasets from Van Kasteren *et* alia. Human activity recognition from wireless sensor network data: Benchmark and software. In Activity Recognition in Pervasive Intelligent Environments. 2011.

What are we looking for again?

- Sensor data is often "big data"
  - Lots of signals being streamed into analytics and/or stored for offline processing
- Analytics tasks
  - Trend analysis
  - Situation/activity recognition
  - Alarms/warnings
- How do we define the "interesting things"?

The instability of classification

- Often a human-centred or -influenced process
  - ...and so subject to changes in behaviour
  - Unintended: different people
  - Unavoidable: assisted living with cognitive decline
  - Intentional: therapeutic interventions
  - Unknown: frequent but unthought-of events
- Implications
  - The classification function may need to change
  - There may be interesting events in the dataset that aren't the subject of a classifier

Unknown event detection

- How do we detect an event for which we haven't explicitly built a classifier?

- Conceptualise the event space as a *mixture model*
  - A sequence of events drawn independently from a set of different distributions
  - May include unknown distributions
  - Does a sensor event fall into a known component, or is it better described by the unknown component?

Defining the sensor readings

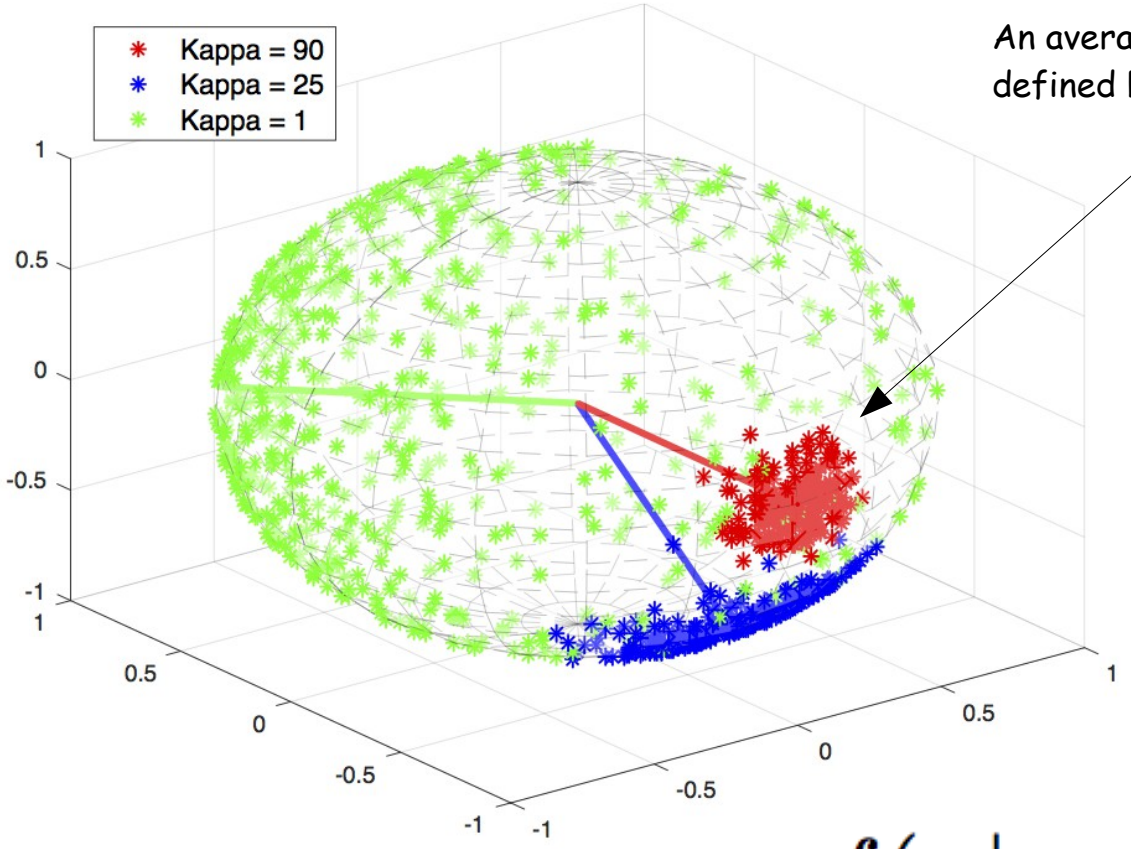
- Give a collection of sensors, at each times slot form a vector $x_S = (x_1, x_2, \ldots, x_S)$ of events
  - A vector in an s-dimensional space of possible readings
  - Can include any sort of sensed feature: continuous, binary, category, time, ...
  - Each dimension has its own semantics

We normalise all the readings, *e.g.*, for a binary sensor we normalise to the proportion of "on" events

- Cluster similar vectors using *cosine distance*
- How would events be distributed?

Distribution of events

- The von Mises Fisher (vMF) model



An average vector and a "spread" defined by the κ parameter

Large values of κ correspond to tightly clustered observations

$$f(\boldsymbol{x}|\boldsymbol{\mu}, \kappa) = c_D(\kappa)e^{\kappa \boldsymbol{\mu}^T \boldsymbol{x}}$$

Mixture model

- Construct a global distribution as:

$$f(\boldsymbol{x}|\Theta) = \sum_{a=1}^{k} \pi_a f_a(\boldsymbol{x}|\Theta^a)$$

Mixture proportions
$\pi_h = p(z = h)$

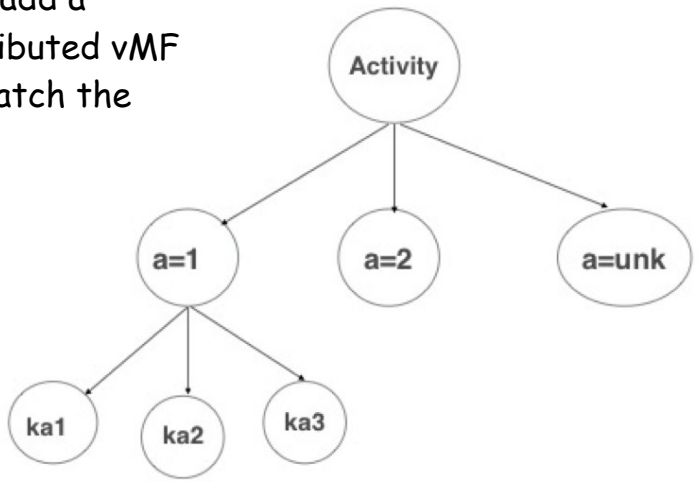
Parameters of the underlying model in the mixture

  - The probability of seeing any given event
  - Identify the model it belongs to

Introducing unknown components

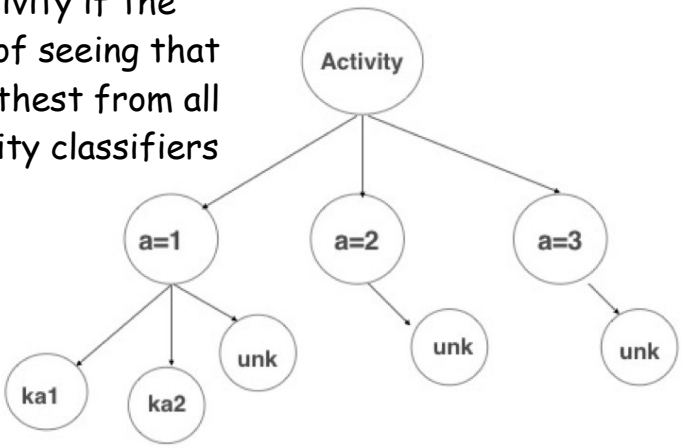
- Two strategies
  - Add a top-level unknown event, the part of the space not covered by existing classifiers
  - For each activity, add an unknown variant that affects the classifier for that activity

In either case, add a uniformly-distributed vMF component to catch the unknown events



(a) strategy 1

An event belongs to an unknown activity if the probability of seeing that event is farthest from all known-activity classifiers



(b) strategy 2

(Very preliminary) results

- Extract an activity from the labelled dataset and see whether the algorithm can find it

# of known activities	HMCIvMFs strategy 1	MvMFs strategy 1	HMCIvMFs strategy 2	MvMFs strategy 2
1	**0.9337 (0.0516)**	0.9007(0.0887)	0.9201(0.0587)	0.8902 (0.0944)
2	0.89 (0.0461)	0.8382 (0.1323)	**0.903 (0.075)**	0.817 (0.0951)
3	0.8151 (0.0552)	0.7738 (0.0877)	**0.895 (0.0497)**	0.7512 (0.0656)
4	0.8115 (0.0771)	0.759 (0.0726)	**0.87 (0.0565)**	0.7731 (0.0913)
5	0.875 (0.053)	0.8132 (0.0505)	**0.8833 (0.043)**	0.73 (0.0771)
6	**0.9203 (0.057)**	0.8478 (0.0711)	0.9132 (0.0537)	0.8225 (0.0899)

- Good separation of unknown events
- Strategy doesn't seem to be critical
- Needs a lot more work....

Fang, Ye, and Dobson. Discovery and recognition of unknown activities by a hierarchical mixture of von Mises Fisher distributions. In preparation for IEEE Trans. KDE.

Discussion

- A very unfamiliar programming environment!
  - Noise convolved onto all the inputs
  - Activities recognised probabilistically
  - Definition of an activity, and the population of activities, may change over time
- No idea how to program systems effectively
  - Move beyond data logging
  - Control decisions
  - Where in the fog?
  - Presenting analysis to users

Three things to take away

- Sensors are becoming ubiquitous, and we need to understand how they affect us
- Statistical machine learning techniques can help pull out the important elements
  - Noise tolerance
  - Multiple targets
  - Unknown events
- Programming is – and will remain – an issue
  - How to express automated decision-making
  - What we do where