University of St Andrews | FOUNDED 1413

Making the transition from sensors to sensor systems software

Simon Dobson
School of Computer Science, University of St Andrews UK

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

DASIP

- What do you talk about here?
  - SoC, manycore architectures, FPGAs, hyperspectral imaging, ...
  - The Square Kilometre Array Computer System
  - Resource Awareness on Heterogeneous MPSoCs for Image Processing

- None of which I can sensibly talk about....

Introduction

- Sensors and imaging often form a (small) part of larger systems
  - Conflicting measurement
  - Exposure to noise, weathering, interference, …
- How do we make decisions and maximise the value of data with these limitations?

- My goal
  - Talk about the transition from sensor to systems

Where??

- 3rd oldest university in the English-speaking world
  - Been teaching computer science since 1969

Science of Sensor Systems Software

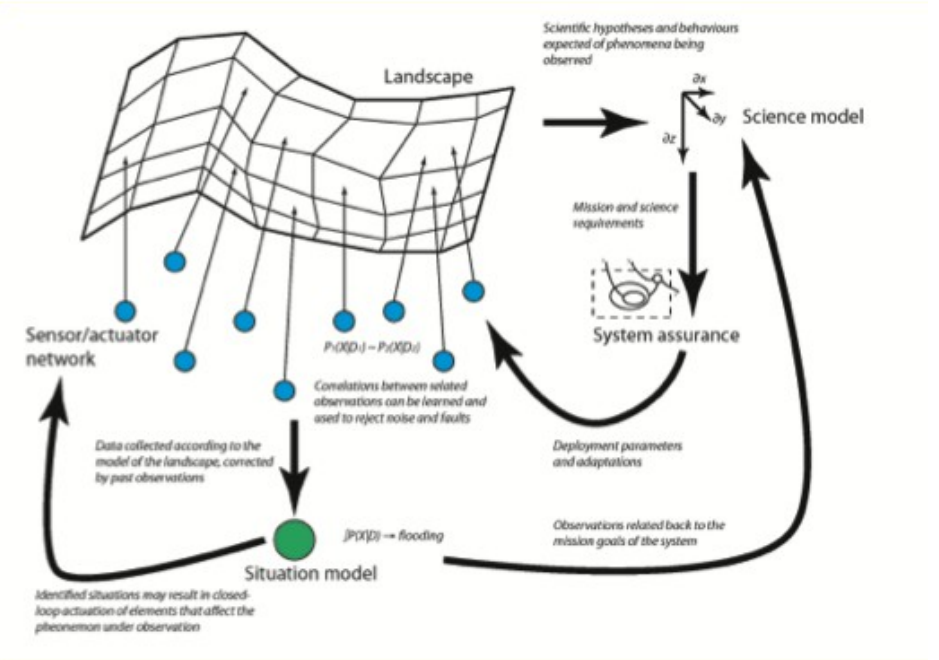
- A five-year, £5.2M EPSRC Programme Grant
  - "Vertical slice" from formal models, through verification and analysis, to deployment
  - Four universities, 10 commercial and agency partners

University of Glasgow

Imperial College London

University of St Andrews

UNIVERSITY OF LIVERPOOL



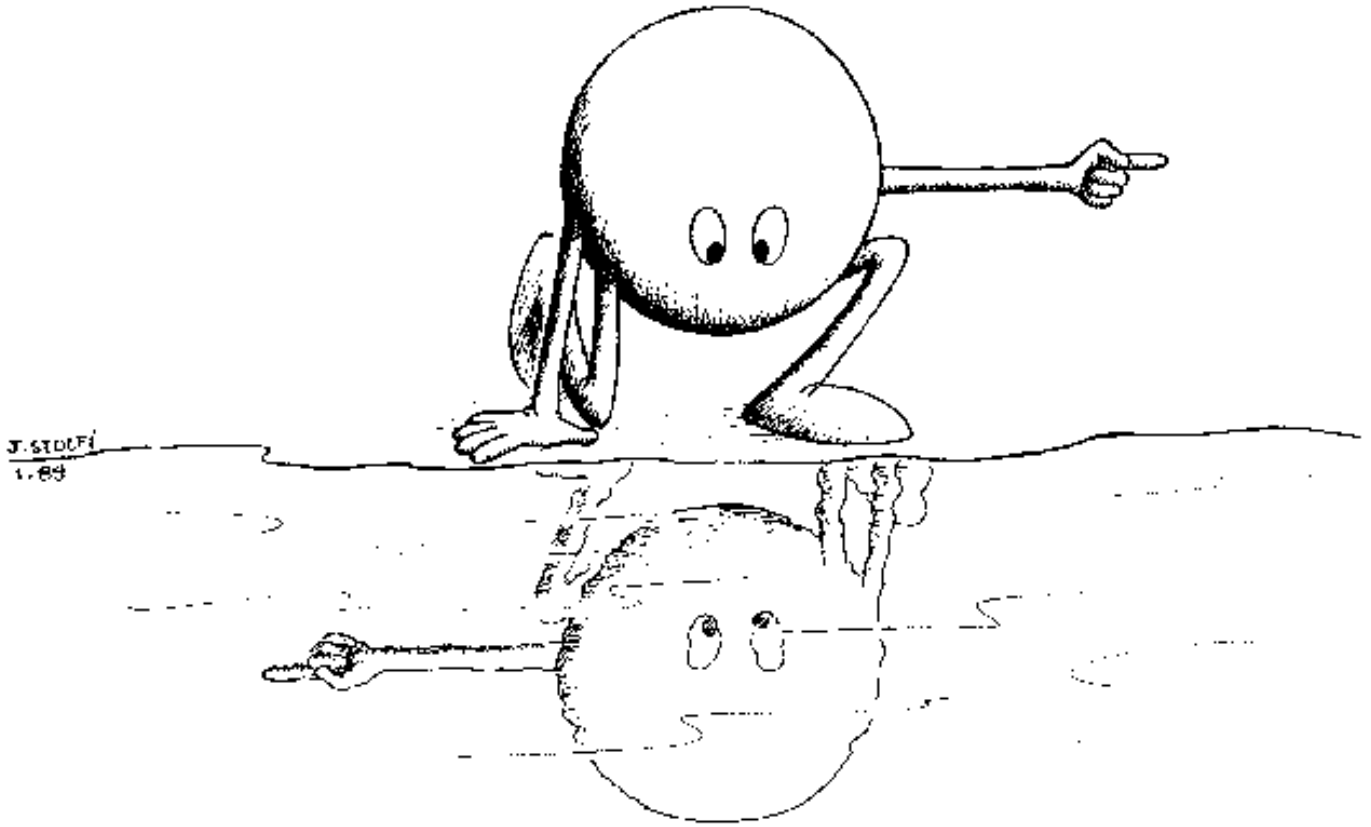
The sensor challenge

- 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

Some of the interesting challenges

- Noise
  - Difficult/impossible to engineer away
  - Deal with *confidence* of the *most probable* signal
- Not knowing what to look for
  - "Unknown" events or anomalies
  - Are there other things going on?
- Long-term storage
  - How do we keep sensor data usable?
  - Applying warehousing techniques to sensor data

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

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

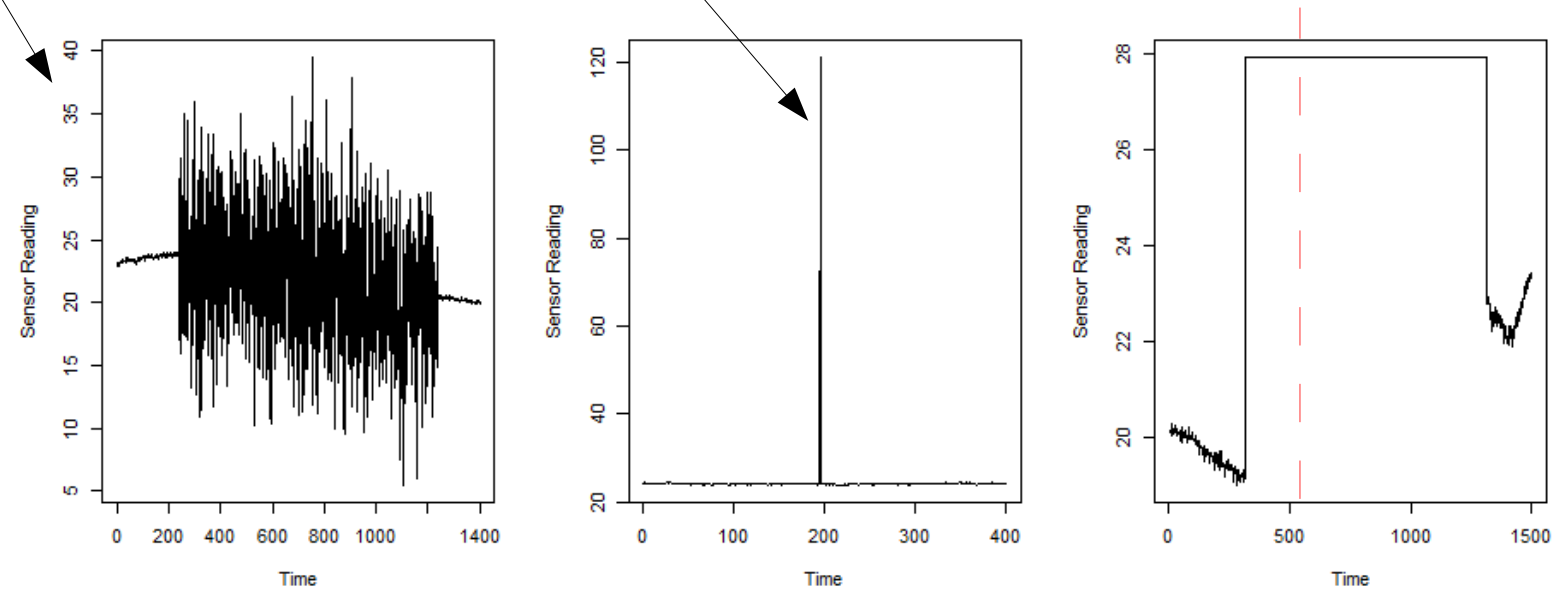
- *Physical* issues that give rise to *faults* in the data
  - Change over time, need autonomic management

Fault types

Natural variation plus noise

Not likely in your data?

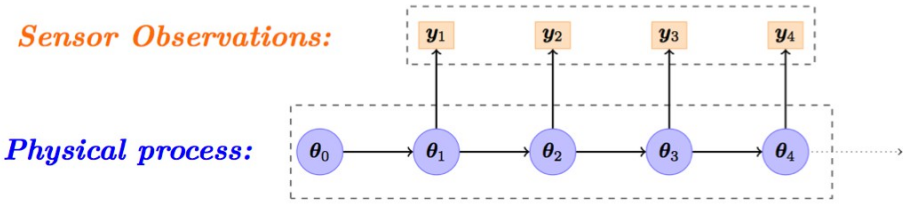
Can we tell here that this is an extended fault? Or is it a change of phenomenon?



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

Approach

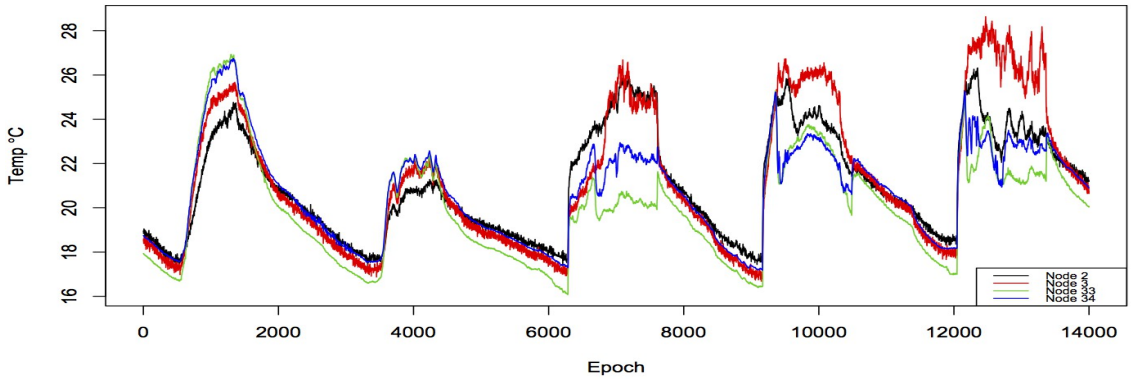
- Re-conceptualise sensors as evidence-providers rather than data- or value-providers
  - Learn the distributions being observed
- Sensor readings *confirm* or *refute* the hypothesis
  - Add *evidence* rather than providing *data*
  - Goal of the network is to *adapt the model* to reflect the conditions being observed on an on-going basis
  - Result is a *distribution* learned from the data
- Correlations let nodes *verify* each other

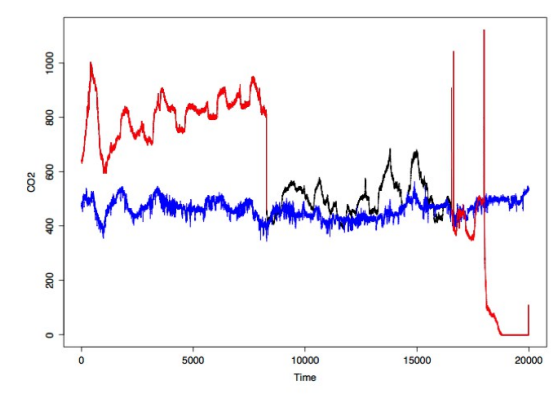


Multi-sensor systems

- Neighbouring nodes observing "the same" phenomenon

Also applies to correlated sensors on different modalities





- 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.

Why Bayesian learning?

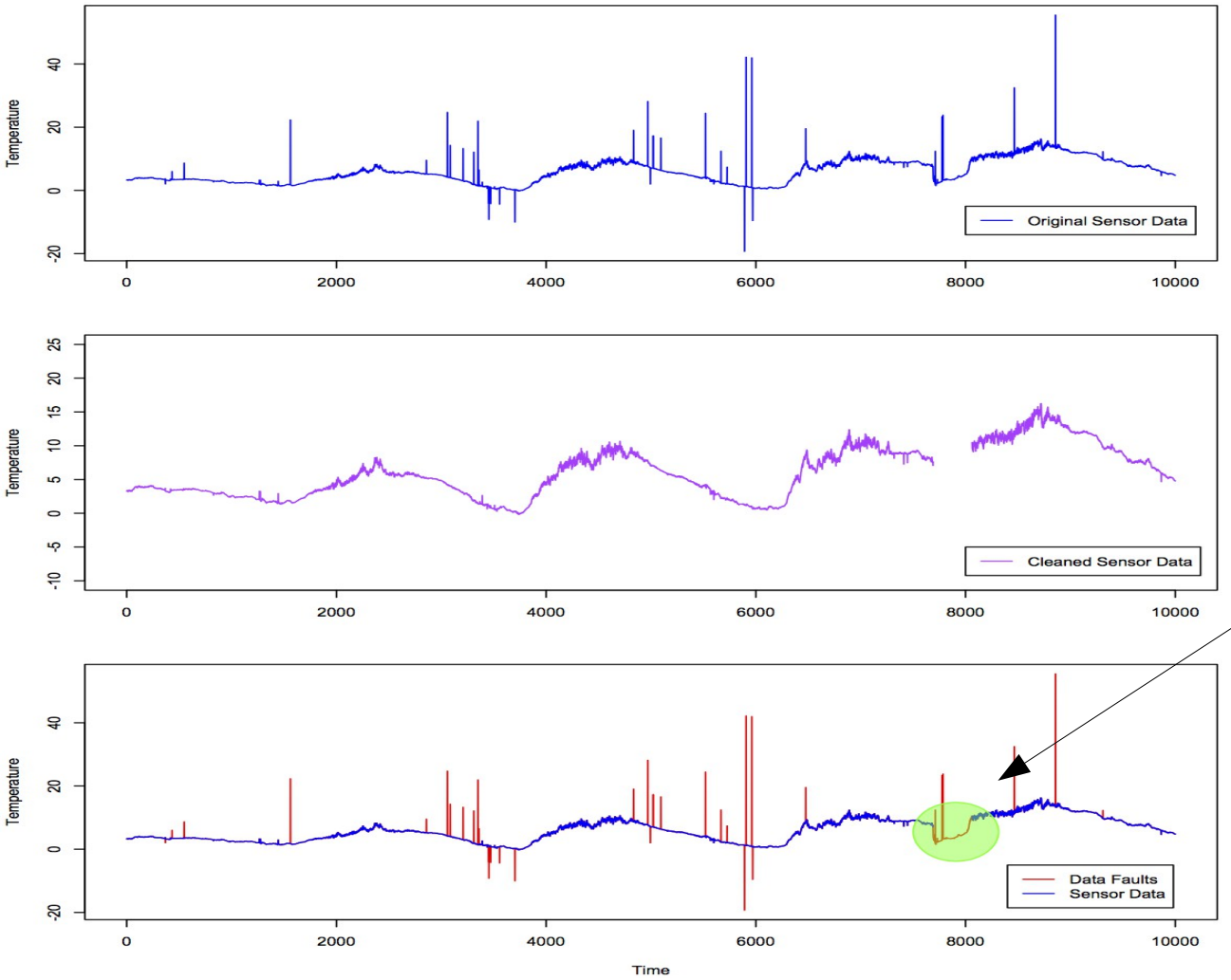
- Formal statistical learning
  - Provides sound inference
- Efficient and lightweight
  - Constant space complexity
  - Constant complexity update
- Robust
  - Test incoming data against the predicted distribution
  - Only then *admit* data for future learning

No need to store any learning data over a protracted period

Example – 1

Original data

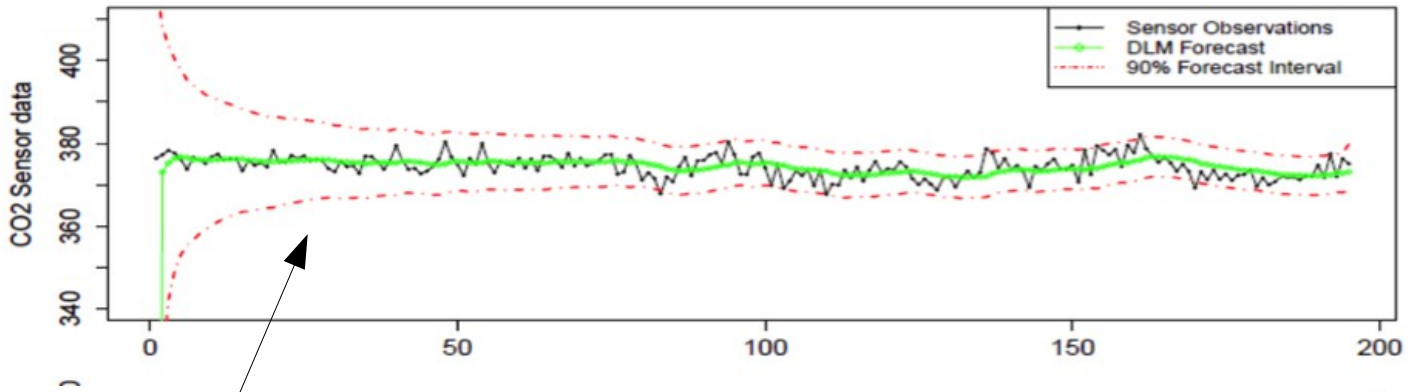
Cleaned data



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

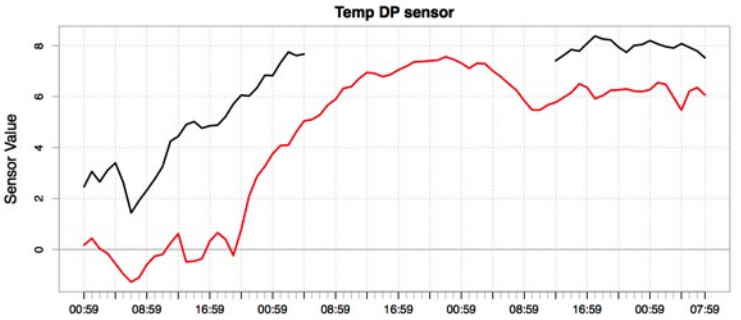
Data from the Lausanne Urban Canopy Experiment (2006)

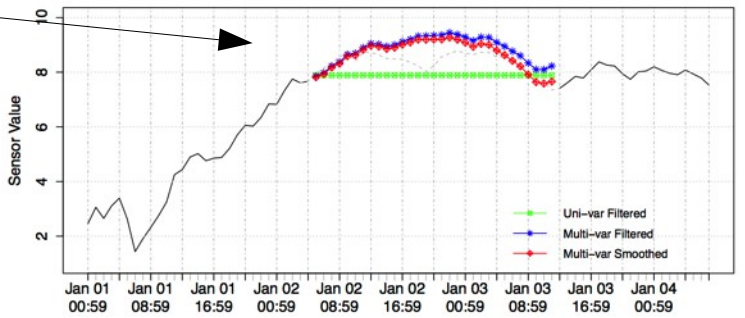
Example – 2



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™

Classifying activities

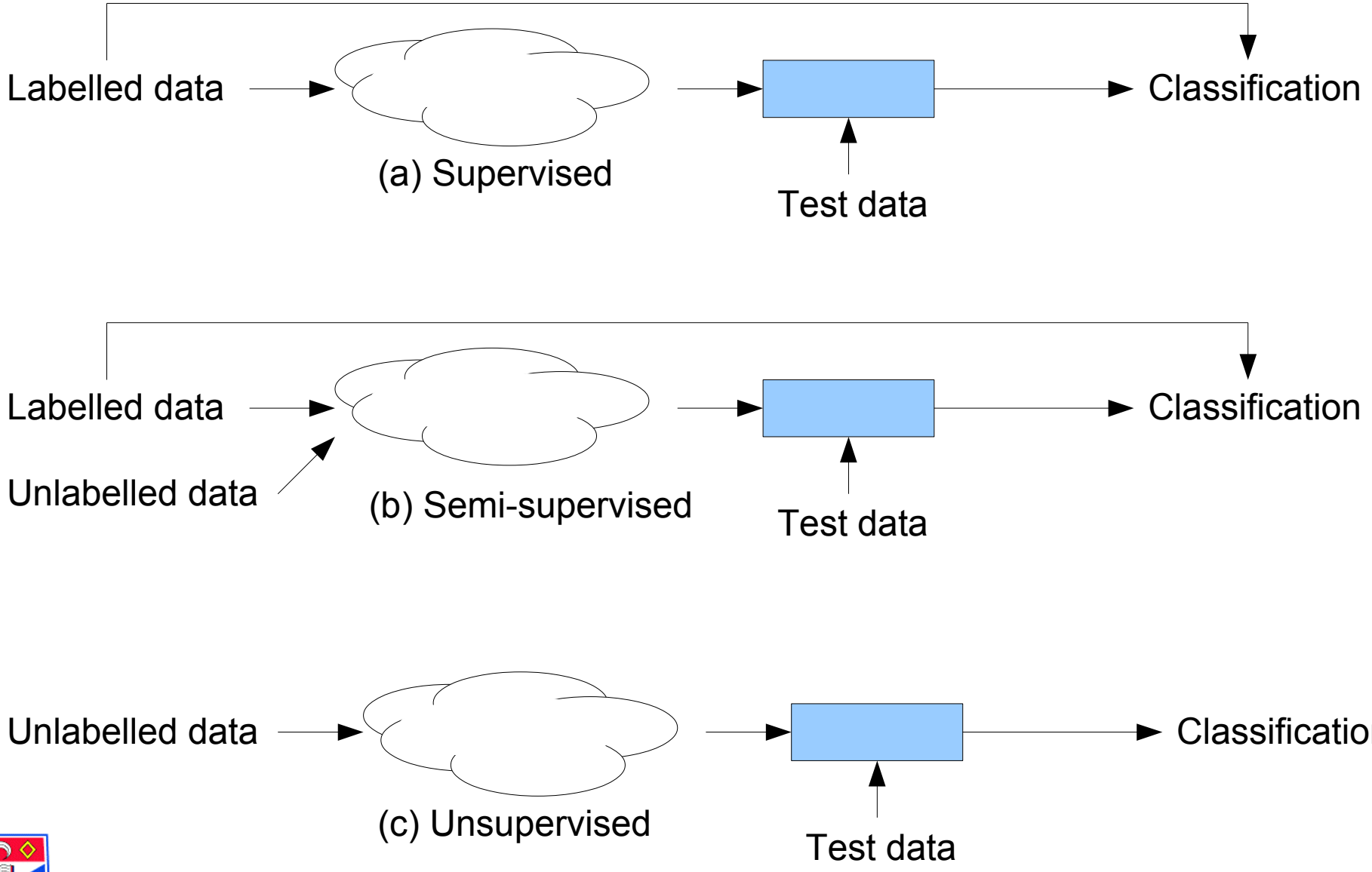
- Often we want sensor input to be *classified*
  - Assigned a human-meaningful interpretation
  - Allows domain experts to understand observations

- The Central Dogma of Machine Learning

The future is like the past

  - (If your system isn't like this, bad things happen)
  - (...and your system often *isn't* like this...)

A brief (yet helpful) guide to ML



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
- Then employ both supervised and semi-supervised approaches to learning events

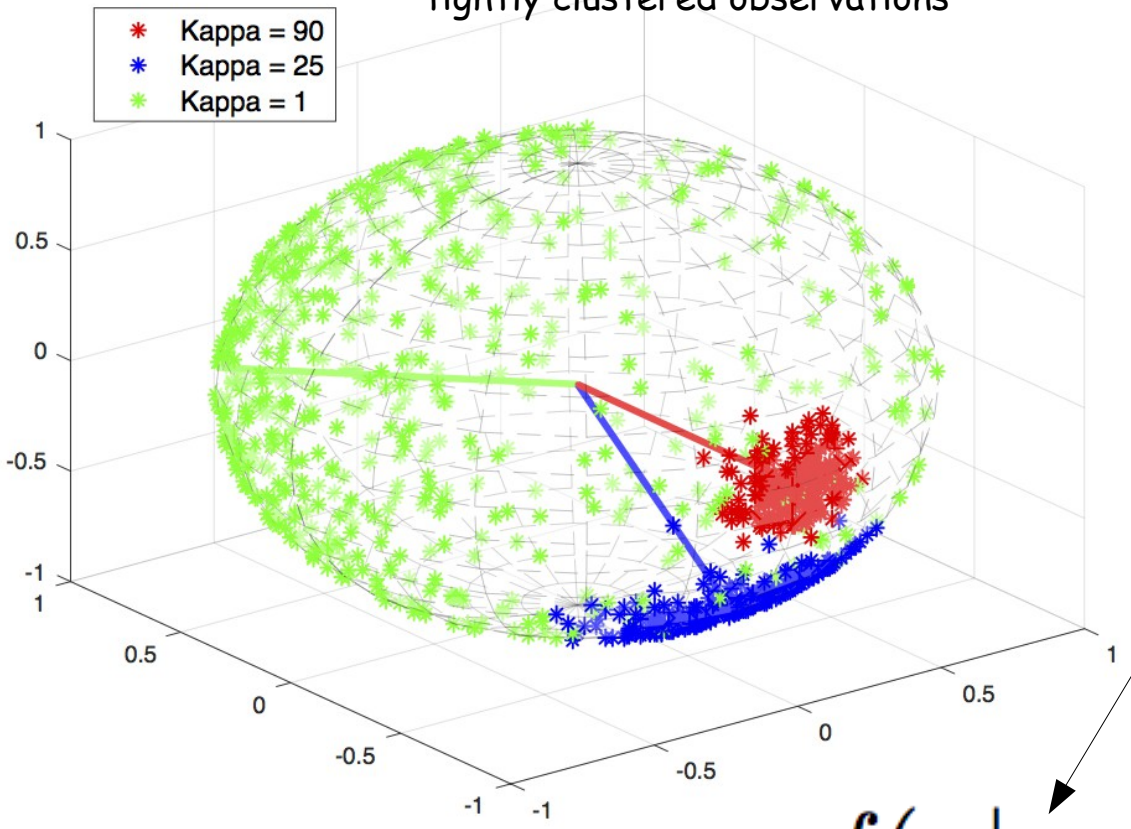
Representing events

- An event is a cluster of similar sensor readings
  - Vectors are "close" in the multidimensional space
  - Different events have different profiles of "closeness"
  - Some are very tight; some more diffuse
- We use directional statistics to capture this
  - Normalise distances to angles: not biased by magnitudes in different dimensions
  - What will the distribution of events look like?

The "normal distribution"

- The von Mises Fisher (vMF) model

Large values of κ correspond to tightly clustered observations



Mean vector

"Variance"

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

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
$\pi_h = p(z = h)$

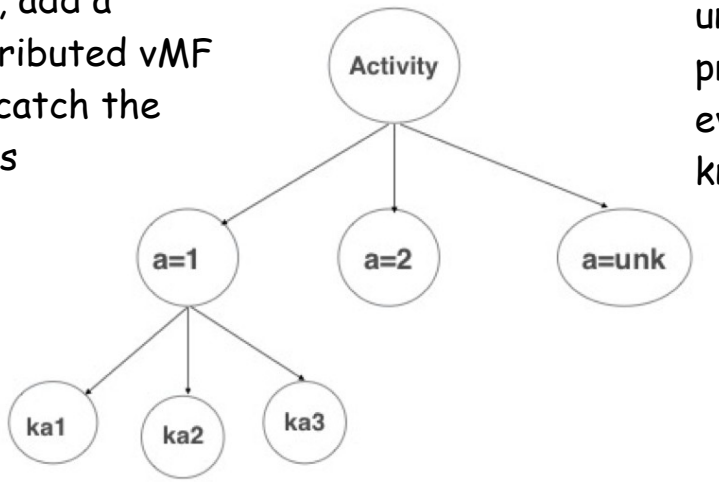
Parameters of the underlying model in the mixture

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

Introducing unknown components

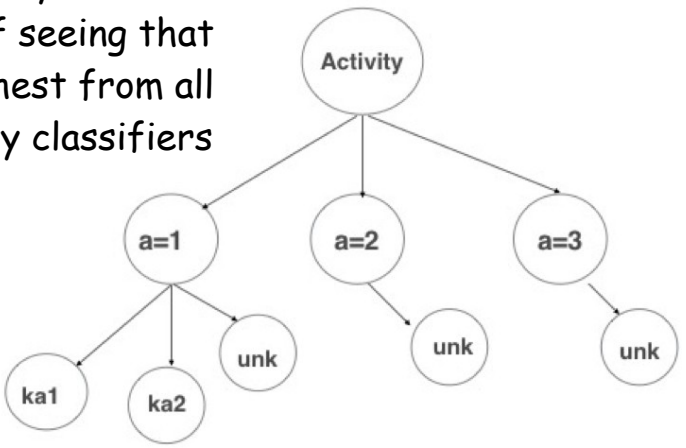
- Two strategies
  - Add a top-level unknown event, the part of the space not covered by existing classifiers
  - For each activity, add an unknown variant that affects the classifier for that activity

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

(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 emerging human activities by hierarchical mixture of directional statistical models. Submitted to IEEE Trans. KDE.

Long-term data value

- Two uses of sensor data
  - Operational: what's happening *now*, for control
  - Archival: what's happened, for analysis

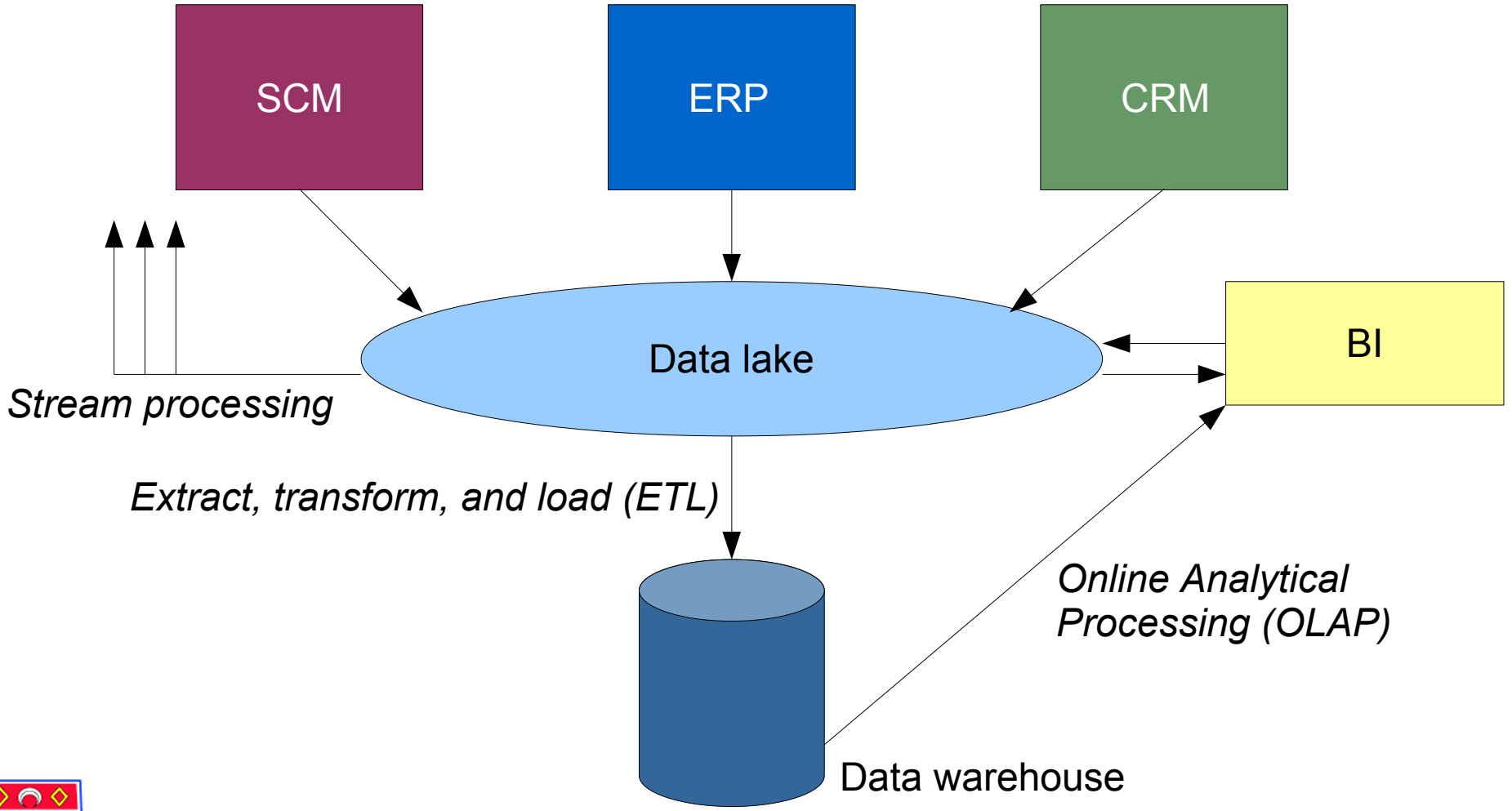
This stage is increasingly valuable for trend analysis, projection, predictive maintenance, and so forth

- These two uses poses different constraints for storage and representation
  - Operational: efficiency, performance, transformations and processing on the fly
  - Archival: interpretability, comparability, stability

How do we support both use cases?

A data warehouse

- Now quite common in data-rich industries



Data warehousing – 1

- Accept that these are two fundamentally different data user cases
  - Operational: data structure, key-value store, …
  - Archival: stable schemata with extensive cross-linkage

Golfarelli and Rizzi. Data warehouse design: modern principles and methodologies. McGraw-Hill. 2009

- The goal of the archive
  - Allow questions across the entire dataset
  - Maintain integrity by storing all the relevant elements – that might operationally be assumed

Data warehousing – 2

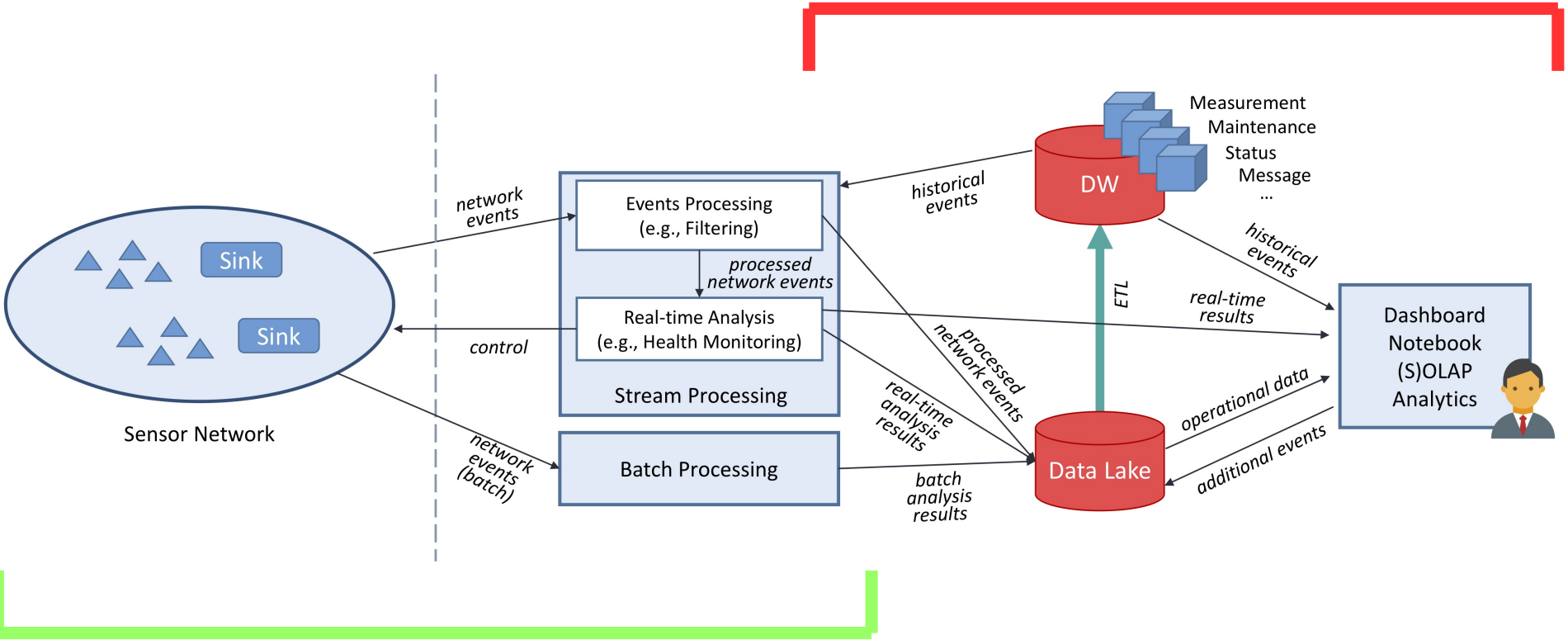
- An explicit *Extract-Transform-Load* (ETL) stage
  - De-operationalise the data ready for archive
  - Regular schema in spite of changes in data lake
  - Works around possibly rapid operation change
  - Archive is never updated after loading

- What happens in the de-operationalising step?
  - Change schemata, add extra elements, ...
  - Things that don't change over operational timescales (but *do* change over archival ones)

Enhanced sensor architecture

Dobson, Golfarelli, Graziani, and Rizzi. A reference architecture and model for sensor data warehousing. IEEE Sensors Journal. 2018.
http://dx.doi.org/10.1109/JSEN.2018.2861327

Data lake and warehousing step

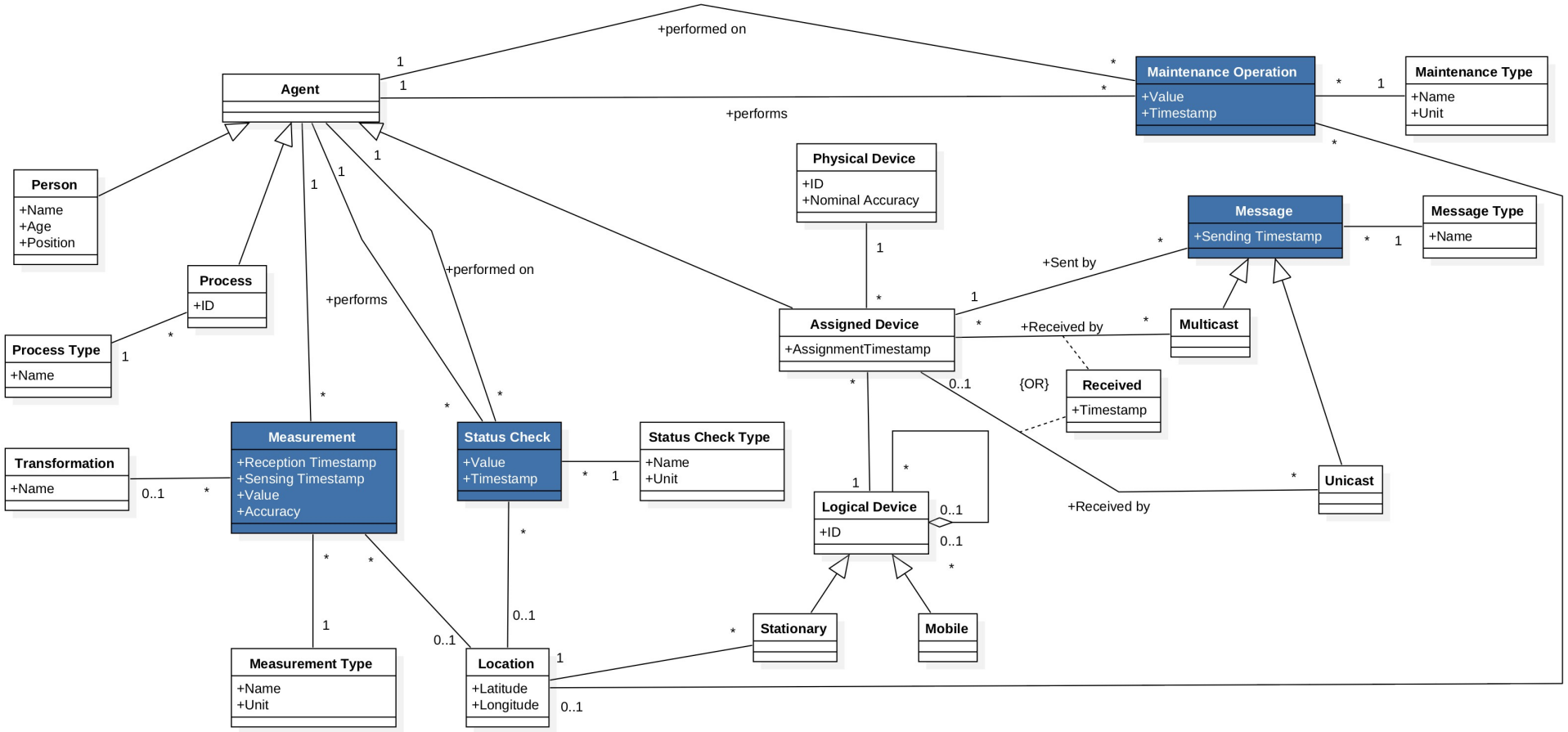


Standard collect-respond-store architecture of whatever kind

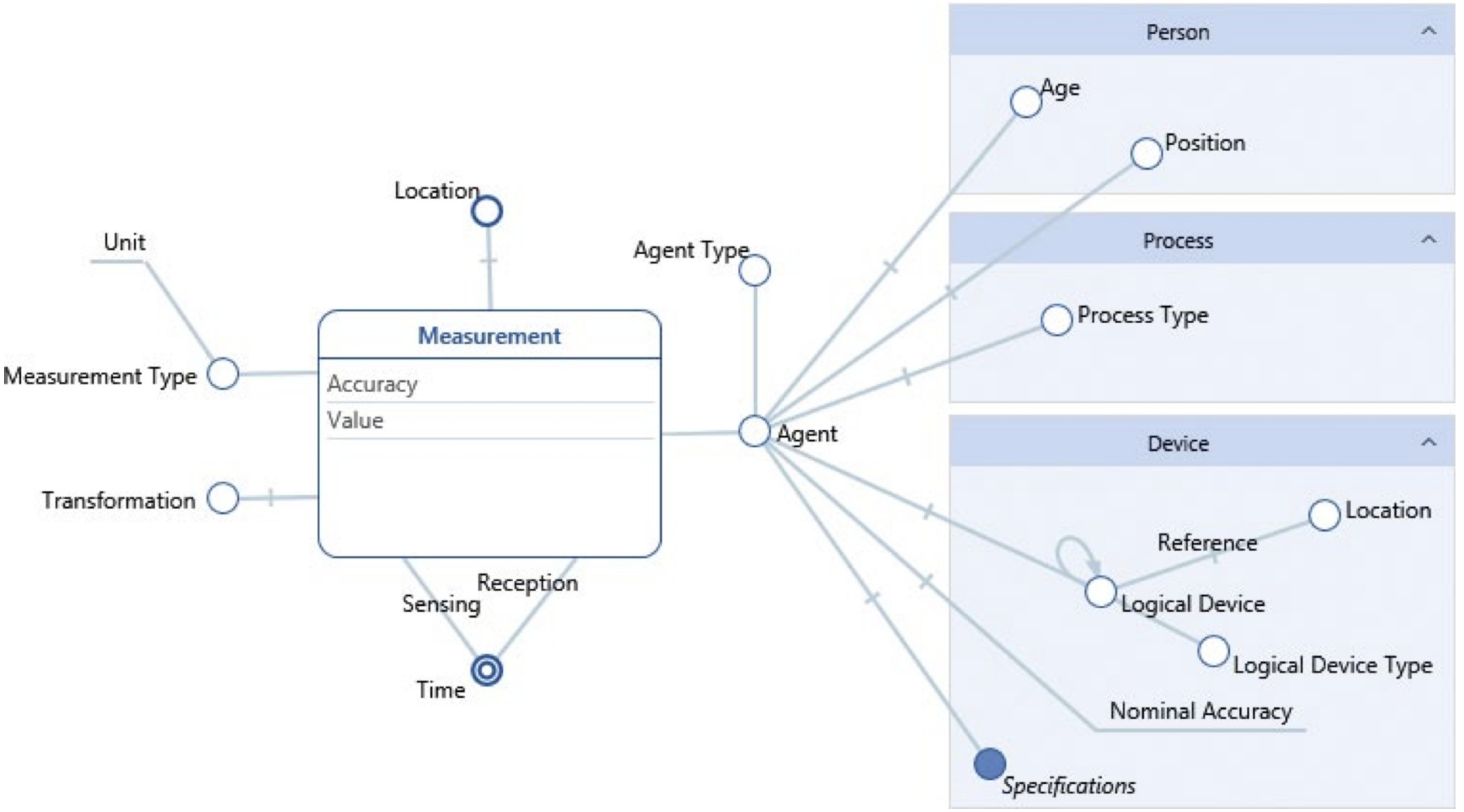
Data analysis

- What do we need to store in order to interpret a dataset *post facto*?
  - The data, obviously
  - The units, any pre- or post-processing
  - The physical characteristics of the sensor, its age, placement
  - The maintenance schedule for the device, since if it's cleaned its data will become “fresher”
  - ...
- A generic warehousing schema for datasets

Classes


Agent
Person
+Name
+Age
+Position
Process
+ID
Process Type
+Name
Transformation
+Name
Measurement
+Reception Timestamp
+Sensing Timestamp
+Value
+Accuracy
Measurement Type
+Name
+Unit
Status Check
+Value
+Timestamp
Status Check Type
+Name
+Unit
Location
+Latitude
+Longitude
Physical Device
+ID
+Nominal Accuracy
Assigned Device
+AssignmentTimestamp
Logical Device
+ID
Stationary
Mobile
Maintenance Operation
+Value
+Timestamp
Maintenance Type
+Name
+Unit
Message
+Sending Timestamp
Message Type
+Name
Multicast
Received
+Timestamp
Unicast
+performed on
+performs
+performed on
+performs
+Sent by
+Received by
{OR}
+Received by
1
*
0..1

Example


Person
Age
Position
Process
Process Type
Device
Location
Reference
Logical Device
Logical Device Type
Nominal Accuracy
Specifications
Location
Unit
Agent Type
Measurement
Accuracy
Value
Measurement Type
Transformation
Agent
Reception
Sensing
Time

Case study: air quality – 1

	Name	Description	Stakeholders	Main Data Sources
(1)	*Air Quality Measurements*	Quantitatively measure the presence of pollutants in the air in proximity of the facility	Facility, Civilians, Agencies	DW (Measurement)*
(2)	*Alerts for High Pollution*	Timely detect the presence of pollutants in quantities exceeding safe thresholds	Facility, Civilians, Agencies	Data Lake
(3)	*Environmental Conditions*	Monitor natural phenomena and properties such as wind, humidity, and temperature in proximity of the facility	Facility, Agencies	DW (Measurement)*
(4)	*Network Health Status*	Detect potentially malfunctioning parts of the network	Facility	Data Lake, DW (Status Check)
(5)	*Maintenance Operations*	Keep track of the maintenance operations executed on the network	Facility, Agencies	DW (Maintenance Operation)

DT AGENT

Agent ID	*Agent*	*Agent Type*	*Position*	*Physical Device*	*Logical Device*	*R. Location*	*Logical Device Type*	*Nominal Accuracy*
1	TH01	Device	-	TH000454746	Rooftop Air Thermometer	Rooftop	Air Thermometer	0.2%
2	TH02	Device	-	TH000932348	Indoor Air Thermometer	Room 01	Air Thermometer	0.2%
3	AN01	Device	-	AN000328434	Rooftop Anemometer	Rooftop	Anemometer	Speed 5%, Direction 4°
4	HY01	Device	-	HY000685347	Rooftop Hygrometer	Rooftop	Air Hygrometer	2%
5	SO01	Device	-	SO000131220	Indoor SO_2 Gas Sensor	Room 01	SO_2 Gas Sensor	15%
6	SO02	Device	-	SO000642337	Rooftop SO_2 Gas Sensor	Rooftop	SO_2 Gas Sensor	15%
7	TE01	Person	Field Engineer	-	-	-	-	-

by *on*

FT MEASUREMENT

S. Time	*R. Time*	*Meas. Type ID*	*Agent ID*	*Transformation*	*Accuracy*	*Value*
2017/10/13 09:00	2017/10/13 09:00	5	5	AVG10	1.2	8
2017/10/13 09:00	2017/10/13 09:00	5	6	AVG10	14.55	97
2017/10/13 09:00	2017/10/13 09:01	1	1	-	0.028	14
2017/10/13 09:00	2017/10/13 09:02	1	2	-	0.04	20
2017/10/13 09:00	2017/10/13 09:02	2	3	-	0.3	6
2017/10/13 09:00	2017/10/13 09:03	3	3	-	4	90
2017/10/13 09:00	2017/10/13 09:03	4	4	-	0.8	40

FT STATUS

Time	*Status Type ID*	*By Agent ID*	*On Agent ID*	*Value*
2017/10/13 09:00	2	5	5	-
2017/10/13 09:01	1	5	5	-
2017/10/13 09:00	2	6	6	-
2017/10/13 09:01	1	6	6	-
2017/10/18 10:00	3	1	1	-

FT MAINTENANCE

Time	*Maint. Type ID*	*By Agent ID*	*On Agent ID*	*Value*
2017/10/20 09:00	1	7	5	80
2017/10/20 09:30	1	7	6	80
2017/10/20 13:00	2	7	5	40
2017/10/26 11:00	3	7	1	20

DT MEASUREMENT TYPE

Type ID	*Type*	*Unit*
1	Temperature	Degrees Celsius
2	Wind Speed	km/h
3	Wind Direction	Degrees Azimuth
4	Humidity	Relative (%)
5	SO_2	$\mu g/m^3$

DT STATUS TYPE

Type ID	*Type*	*Unit*
1	Asleep	None
2	Active	None
3	Battery Exhausted	None

DT MAINTENANCE TYPE

Type ID	*Type*	*Unit*
1	Calibration	Cost (Euro)
2	Filter Replacement	Cost (Euro)
3	Battery Replacement	Cost (Euro)

Use cases taken from our work with Topolytics at the Grangemouth facility

Case study: air quality – 2

- Support drill-down by combining data from across operational datasets
  - Unlikely to be linked at the operational level
  - Different stakeholders
  - Whole-system view?

Weekly SO_2 Levels

Week	*Location*	SO_2	*Wind (km/h)*	*Temperature*
2017/10/02 – 2017/10/08	Rooftop	98	7	13
	Room 01	11	-	20
2017/10/09 – 2017/10/15	Rooftop	105	9	15
	Room 01	13	-	20
2017/10/16 – 2017/10/22	Rooftop	144	4	15
	Room 01	16	-	20

drill-down

Daily SO_2 Levels for Week 2017/10/16 and Rooftop SO_2 Gas Sensor

Date	SO_2	*Days Since Last Calibration*
2017/10/16	135	15
2017/10/17	170	16
2017/10/18	192	17
2017/10/19	205	18
2017/10/20	106	0
2017/10/21	102	1
2017/10/22	98	2

We've observed similar fragmentation in other industrial settings

Three things to take away

- We engineer systems around sensors
  - Providing *evidence of value*, not values themselves
- Statistical machine learning techniques can help pull important elements out of sensed data
- Data has a lifecycle that needs to be supported – and there are known techniques for this
  - Can we do things in the devices to maximise the long-term value they generate?