MANCHESTER 1824

The University of Manchester

# Awareness Requirements & the Implications of Change

Alistair Sutcliffe

University College, London &
University of Manchester

Trento Oct 2011

# Objectives

1. Taxonomise the sources of change-

   - where and what to monitor- awareness requirements

2. Analyse the nature of change

   - how to monitor and interpret change

3. Investigate the implications

   - adaptation strategies and trade offs

# Presentation outline

1. Sources of AR- lessons from safety critical literature
2. Some examples from London Ambulance service case study
3. Classes of Awareness requirements (monitor processes)
4. Implications for change- (adaptation strategies)
5. Lesson from aviation case studies
6. Implications & future work

# Background/sources

- RE self adaptive systems
  - Awareness requirements (Mylopoulos, Souza et al 2011)
  - ReqMon & EEAT (Robinson, Fickas)
  - RELAX requirements adaptation (Sawyer, Whittle et al)
  - Self aware systems (Ghezzi)

- Safety Critical Systems
  - failure causation analysis (Hollnagel, Johnson, Leveson)
  - human error theory (Reason, Woods)

- Safety critical RE & Generic RE models
  - scenario analysis, PCRE (Sutcliffe et al 1999, 2005)
  - domain theory (Sutcliffe 2002)

# Awareness Requirements - fundamental types

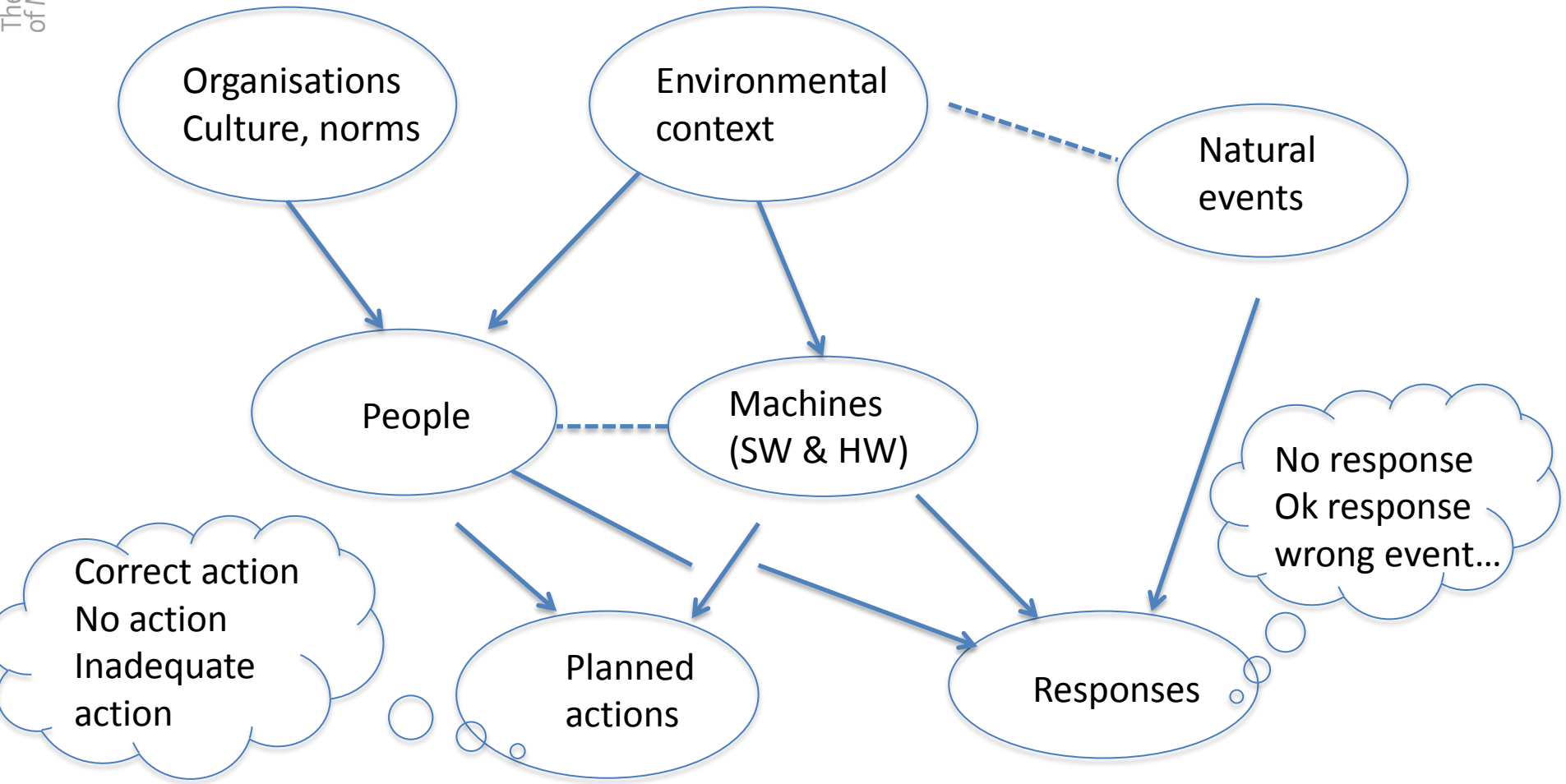
1. Event (failure) awareness

- Safety critical, command and control, automated systems
- ARs are integrated into the RE process
  - Functional requirements for normal goals
  - Functional requirements for exceptions, alternative paths etc
- objective is to deal with exceptions and unexpected events

2. Performance- Level of Service awareness

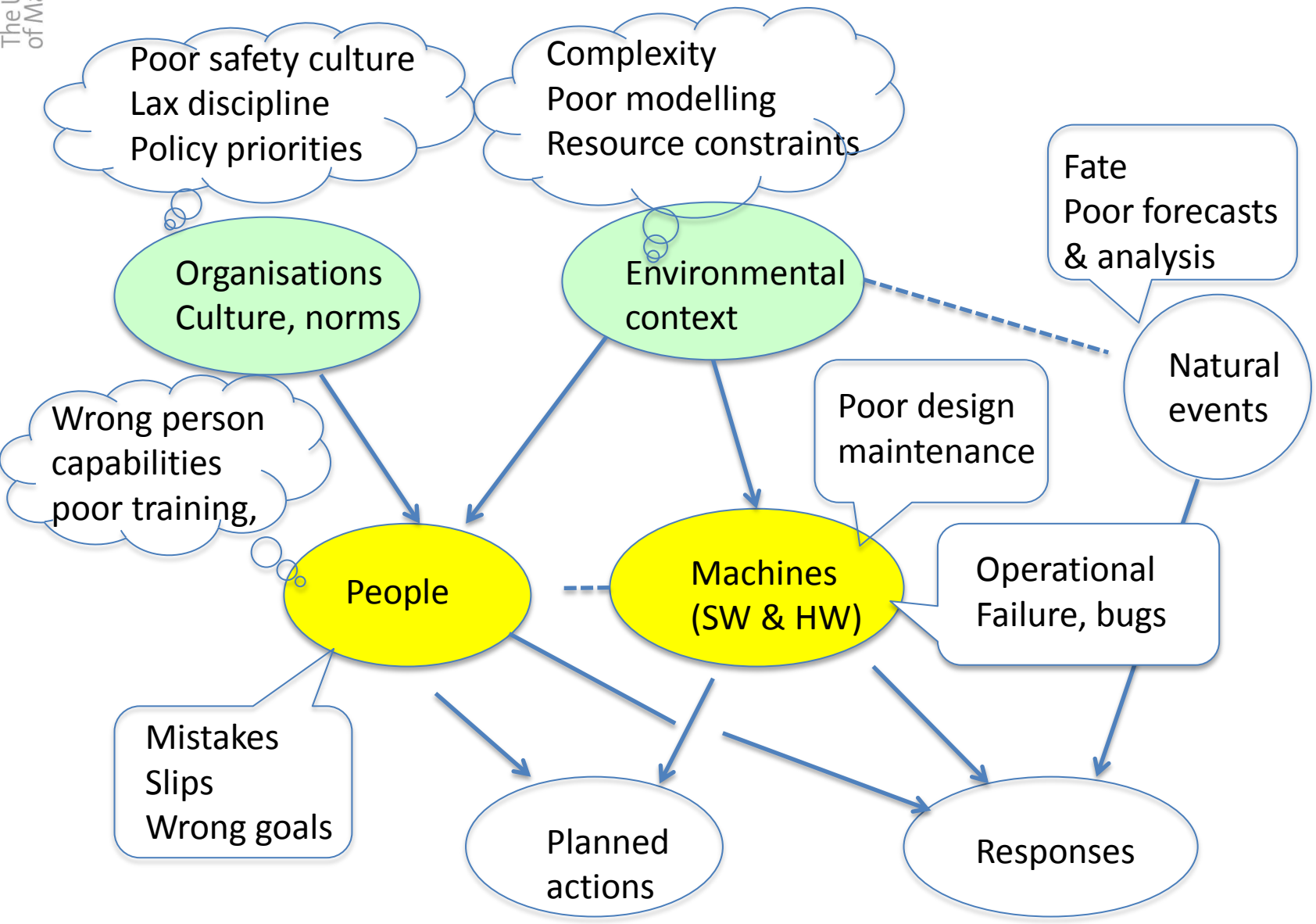
- ARs are supplementary to normal Requirements
  - requirements for monitors and adaptive processes
- objective is to tune/improve the current system, or adapt to contextual changes

# Sources of Failure




Want more detail ?
Johnson W. (1980), Management Oversight Risk Tree (MORT), US Dept of Energy report
10,000 nodes in generic failure diagnosis tree

# Sources of Failure- some causes

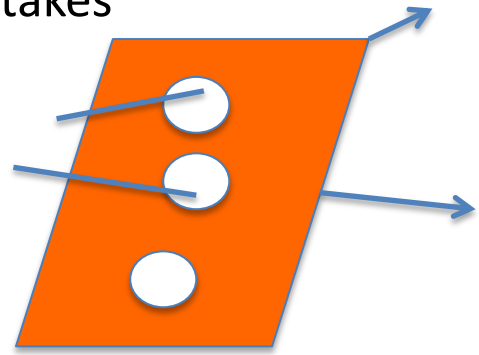


# Causes of Failure

- Indirect causes- preconditions or states that allow undesired events to happen
  - poor policies and goals
  - culture and norms
  - complex and unpredictable environments

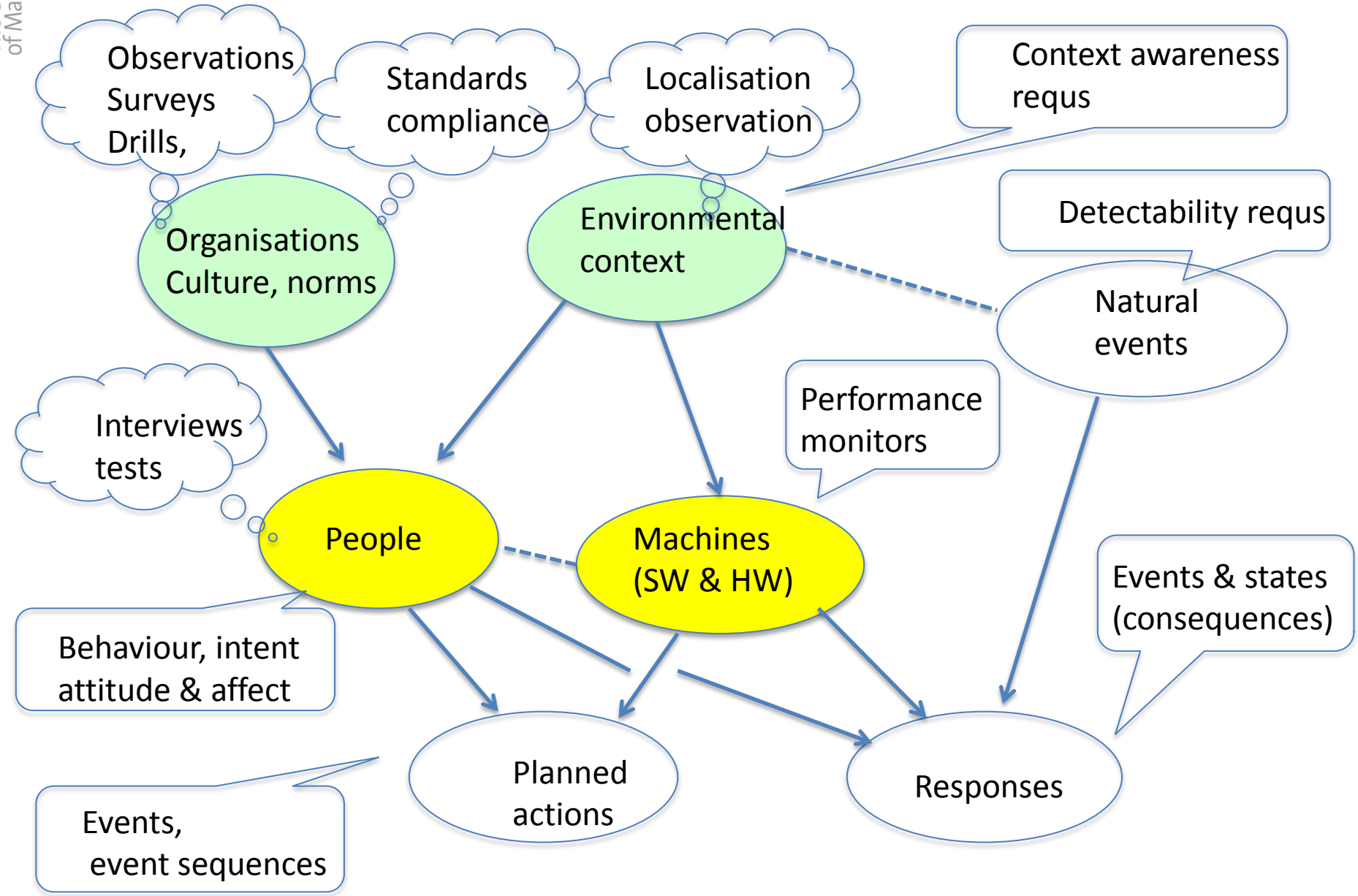
- Direct causes- failures by people or machines
  - Errors in planned procedures bugs, slips, lapses
  - Design failures unexpected events, incorrect response planned
  - Poor decisions, mistakes

Opportunities for unexpected or dangerous events



Reason's 'Swiss cheese model'

# Monitoring Methods

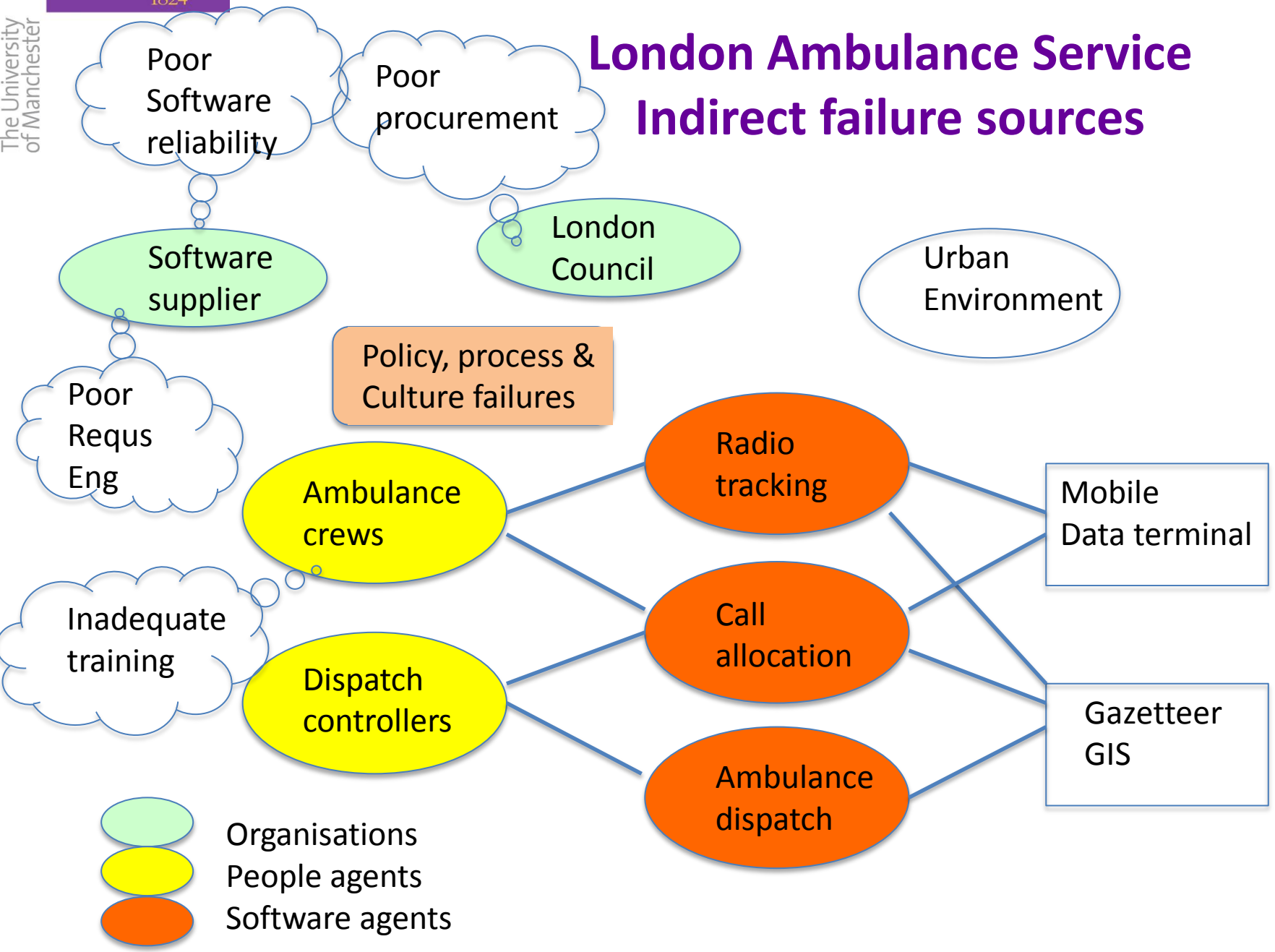


# AR Types (Monitors)

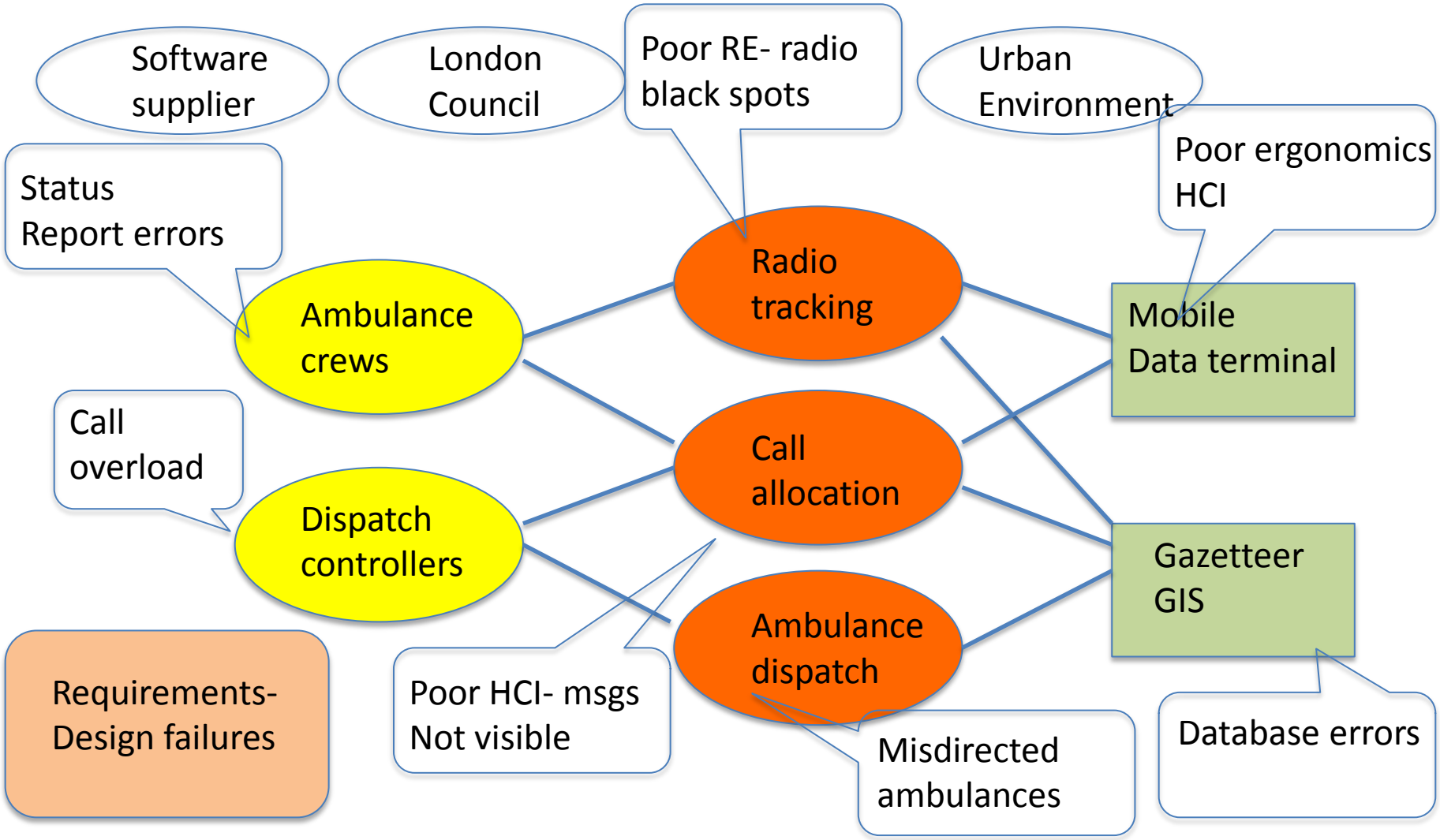
- Soft Monitors- Awareness requirements which can only by captured indirectly by people
  - by observation, interviews
  - surveys
  - standards compliance, certification
  - running tests, drills to check system performance
  - decision support analysis tools (e.g. statistical tests)

- Hard Monitors- Awareness requirements which can be captured automatically (or set as thresholds, targets, indicators, etc)
  - simple event analysers
  - compound event analysers- sequences, cumulative events
  - context analysers- event and states
  - complex event analysers, data miners with history

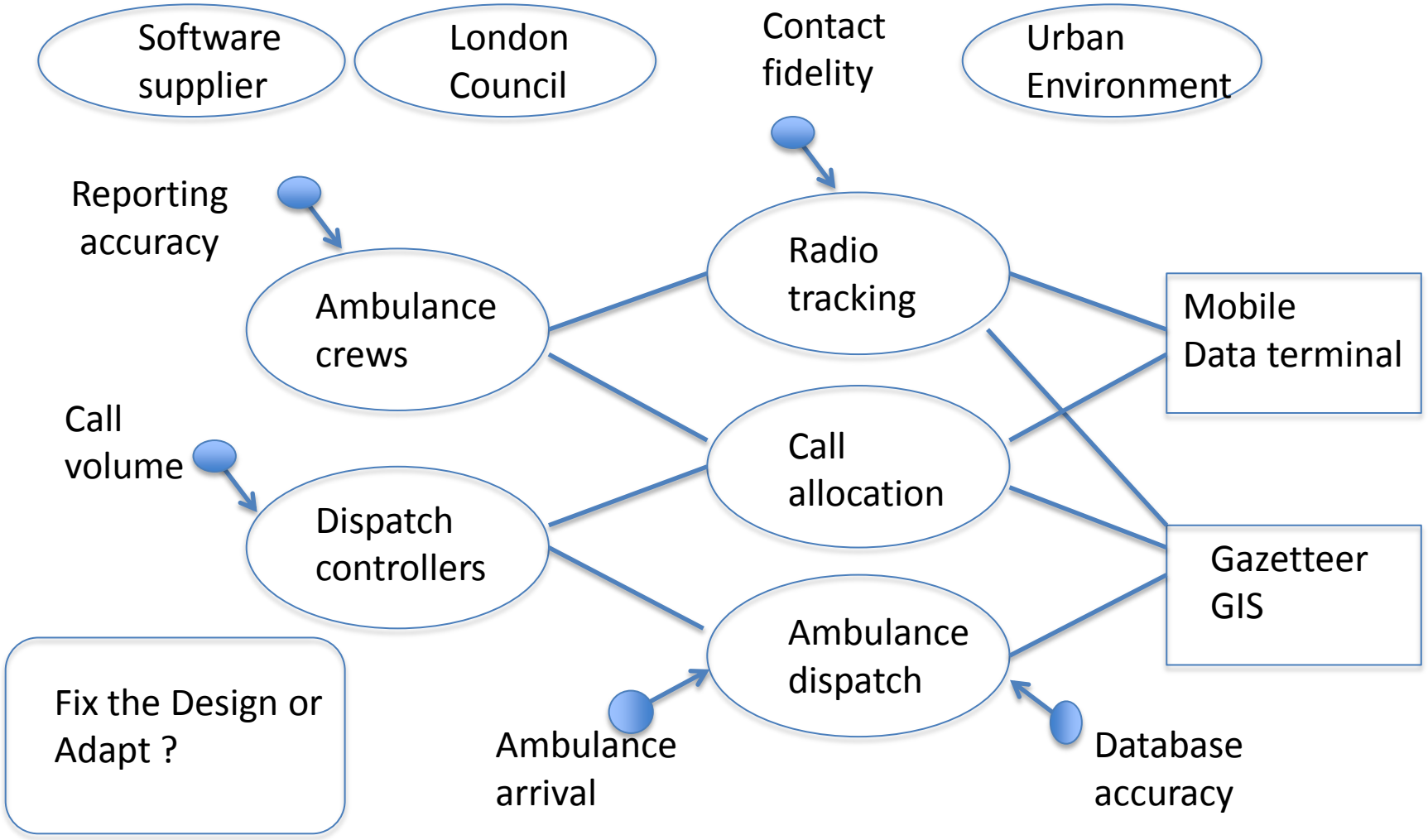
# London Ambulance Service
## Indirect failure sources



# LAS Direct Failure Sources



# LAS Possible ARs



# Analysing ARs

An arbitrary number of ARs and adaptive process could be specified but

we need a systematic process to:

i. Identify ARs that are a a necessary part of the problem domain (not Reqs, design errors)

ii. Elicit and analyse the sufficient and necessary set of ARs

iii. Plan appropriate adaptations

# AR Methodology (starting points)

- Framework of problem domains

  - context aware, location aware applications
  - mobile applications
  - customisable and configurable systems
  - short term and long term adaptive systems

- Type theory of ARs, what to go looking for at the

  event level

  performance level

- Adaptation strategies (linked to AR types)

# Awareness Requirements Types 1.

- Agent (People) Monitors
  -monitoring states/ properties of agents,
    e.g. health care blood pressure, body temperature
  - monitoring agent behaviour
    e.g. heart rate, respiratory rate, gestures, movement
  -monitoring intent and emotional state
    e.g. stress by heart rate and GSR,
      intent from behaviour, analysing computer operation in email
      (see PCRE personal goals Sutcliffe et al 2005)
  -performance monitors
    e.g. exercise routines, calories burned, aerobic exercise level

# Awareness Requirements Types 2

- Artefact (machine and environment monitors)

  - environment state, e.g. temperature, luminance, noise

  - artefact state cf Required Behaviour in Problem frames, e.g. door open/ closed.

  - artefact state in the world,
    location in space, 2D or 3D coordinates, GPS tracking
    location within a reference model, locus on map, on pathway, etc

  - artefact behaviour
    actions compared with plan
    response to events

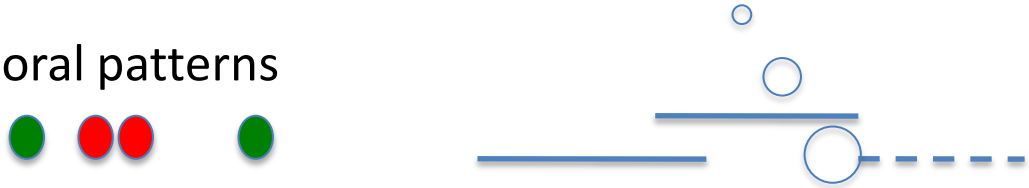
# Awareness Requirements Types 3 state/event monitors

- State value, discrete, continuous, boolean

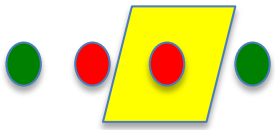
- Event identity

- Event patterns

- Temporal patterns



- Event –state monitors

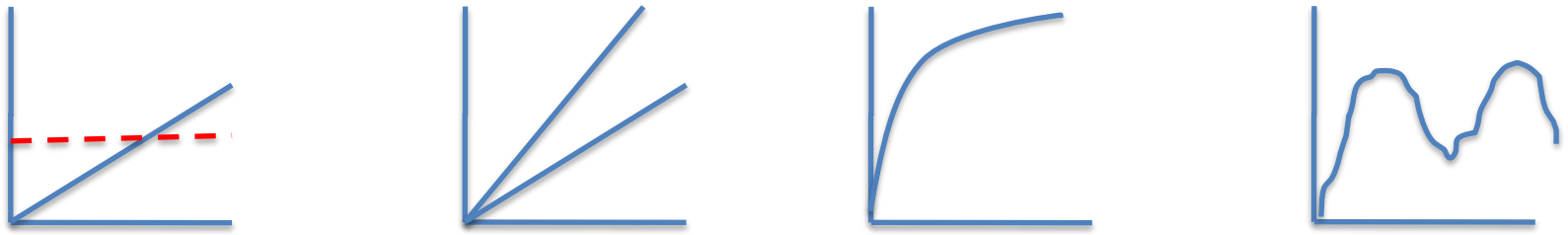


For an event pattern taxonomy
See Hollnagel (1999)
CREAM

# Awareness Requirements 4 performance monitors

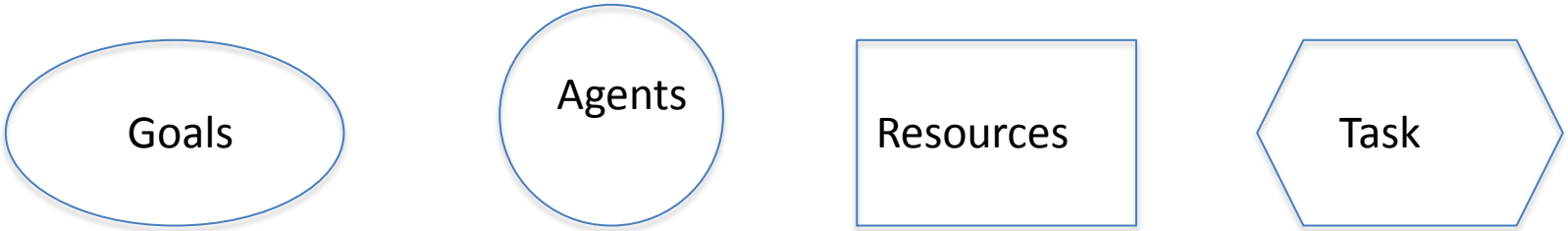
- Aggregate data from event level monitors
  - over time
  - across individuals
  - classify events, categories, distributions

- Compare aggregated data against a target (threshold, indicator) or for desired patterns



# Analysis Process

1. Walkthrough model (i* or take your pick), identify sources of change
2. Inquire which type of ARs are appropriate/needed by component
3. Specify ARs as Monitors /Sensors
4. Specify Interpreters if necessary (performance ARs)




Performance
ARs, indicators
Attained state AR4

People
Artefact
AR type 1 & 2

{environment}
Artefact
AR type 2

Event
AR type 3

# Implications for change (adaptation strategies)

Safety Critical ARs event level

Goal: to adapt quickly or whole system fails

- instance level
  - repeat action (retry after interval)
  - use default value/ setting
  - use history repeat last successful action

- goal/method level
  - select alternative rule/ method
  - backtrack and use previous (successful) method
  - delegate to human intervention

# Adaptation strategies (performance level)

Goal: To improve performance towards desired /observed goal or level of service.

(i) Performance tuning- go faster, more slowly, run more often.. Etc
run time controls (arrange more meetings)

(ii) Relax constraints (N people in room, meeting time, people invited)

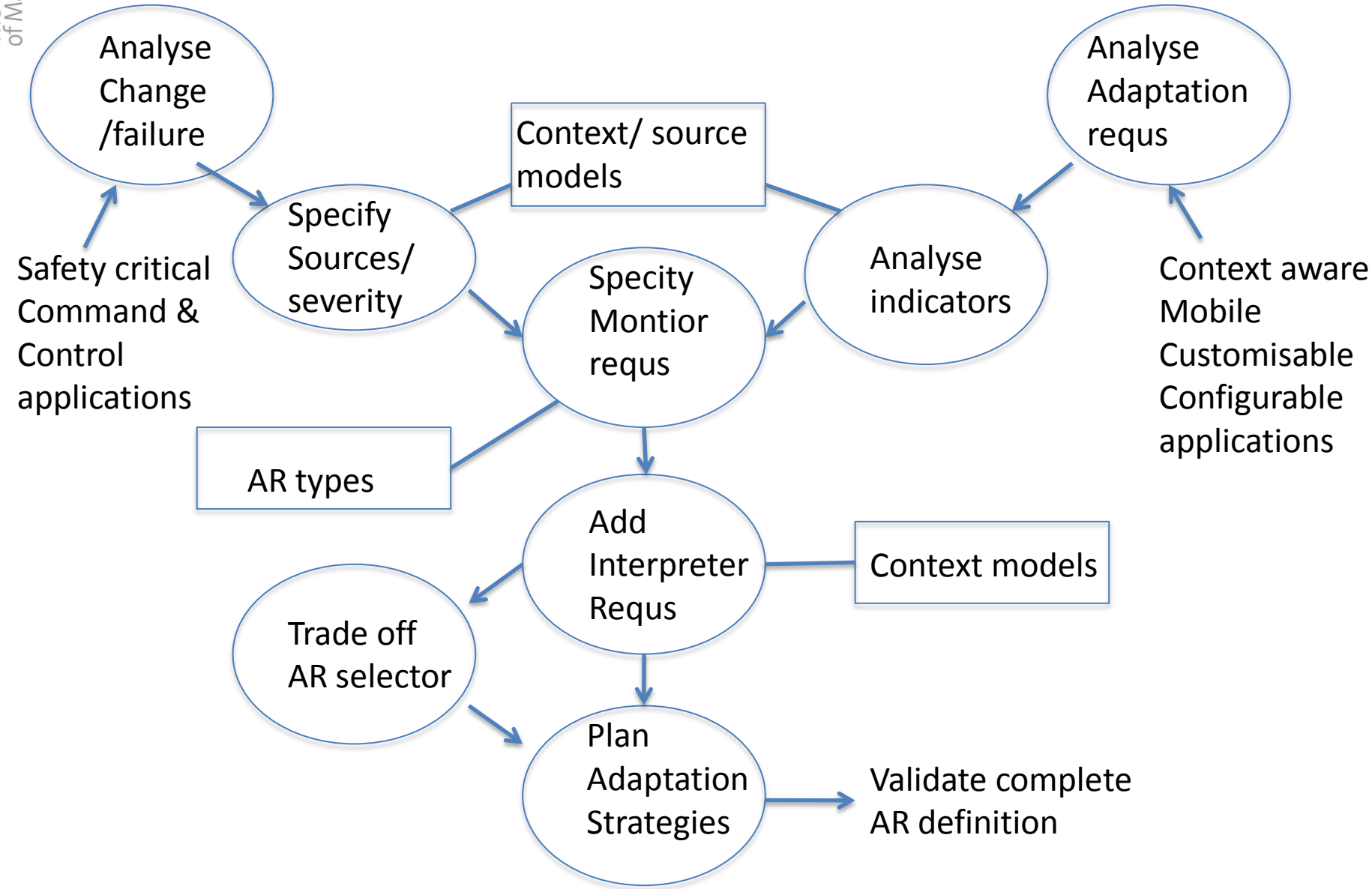
(iii) Adapt resources (larger room, more locations)

(iv) Change process (use Doodle web meeting scheduler)

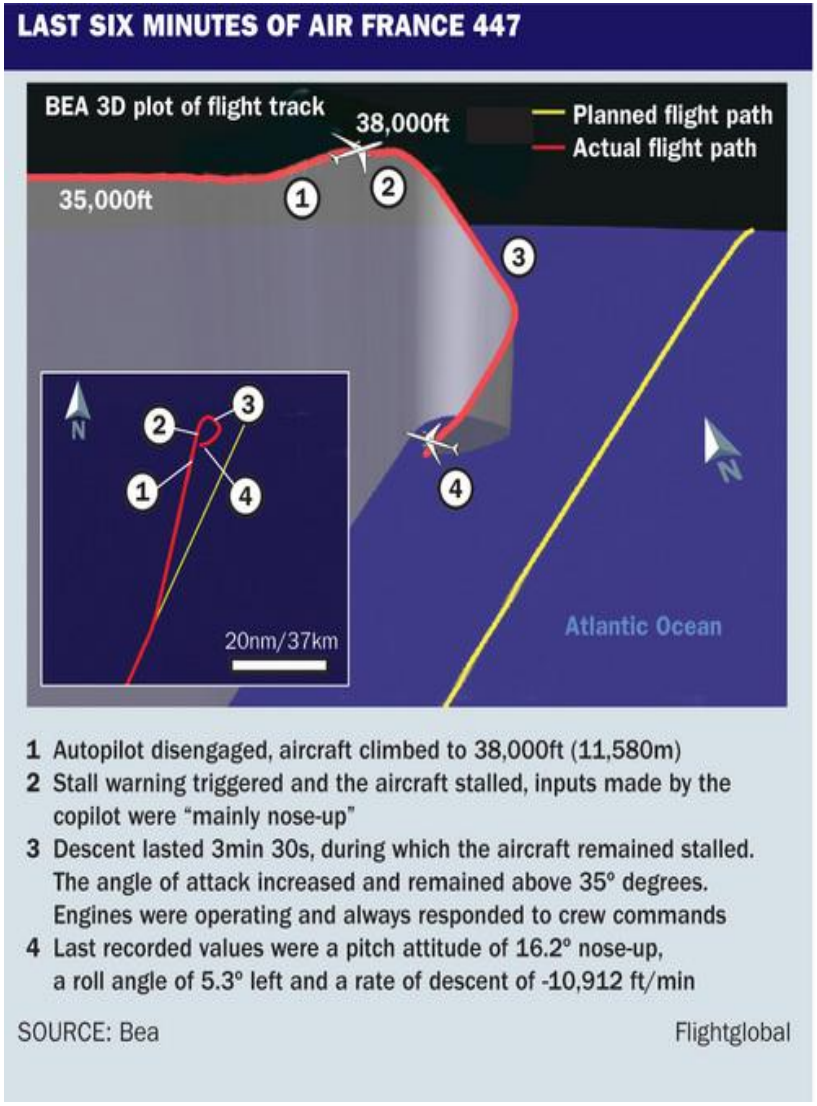
(v) Change method/algorithm (best fit, approximate fit, video conference)

# Process- AR Specification

## (2 routes method)



# Awareness Requs
# lessons from aviation

# Awareness Requs

## lessons from aviation

Air France Airbus 330 South Atlantic 2010

Flying on autopilot when aircraft encountered a storm. Pilot probe sensor for airspeed froze and stopping functioning.

This caused the autopilot to trip (no airspeed can't fly plane, so delegate control to the pilot).

NB Meta Awareness Requirement strategy: if sensor fails then can not adapt, so delegate

The pilots had no warning of the failure and the aircraft was flying at 38,000'

at this altitude jet aircraft can easily stall…….

# Causal Analysis

The pilot probe problem was known- they were fixing the design but hadn't changed it on this aircraft-

safety culture- policy failure

The problem of high altitude stall is known and so is the cure- throttle up and dive 5%, but pilots are rarely trained in simulators for stall recovery

policy, training and procedural knowledge error

Could the design (awareness requirements) have fixed the problem ?

# Possible fixes

Awareness Requ rule

If

Altitude > 37,000’ AND airspeed <530 mph AND autopliot trip

Then

Increase throttle 15%, dive 5% for 10 secs

Alert Pilot

Hand over control after 10 secs

But this is with 20/20 hind sight

And could this adaptation be dangerous in the future ?

{try the rule in dense air traffic}

# Lesson from Aviation 2

Lufthansa Airbus 320 landing at Warsaw airport 1996

Heavy rain at the time, and a strong cross wind

Pilots opted to land manually, landed left undercarriage first then

- Applied the brakes- nothing happened !
- Tried thrust reversers- nothing happened !
- Panick ! Too late to go around…..

Made of mess of the lights at the end of the runway- AR design failure

NB: it was normal (but not officially advised) flying practice to land on one undercarriage leg in a cross wind. The Requirements Engineers never interviewed pilots.

# Causal Analysis

The automated flight management systems had an Awareness Requirement rule

IF

Both undercarriage legs are in contact with the ground AND wheels turning

THEN

Enable Thrust Reversers

It was there for a good reason: Air Lauder Boeing 757 had accidently engaged thrust reversers in flight- not advised.

BUT state-event interaction in Awareness Requirements

20/20 foresight needed to anticipate future system states, when combination of states and events approaches infinity

# Some lessons
# Awareness Requs @ the event level

- Awareness requirements and automated adaptation can be dangerous
- Adaptation in one state may be safe but you can't anticipate (or monitor) all future states
- RE challenges for Awareness Requirements @ the event level

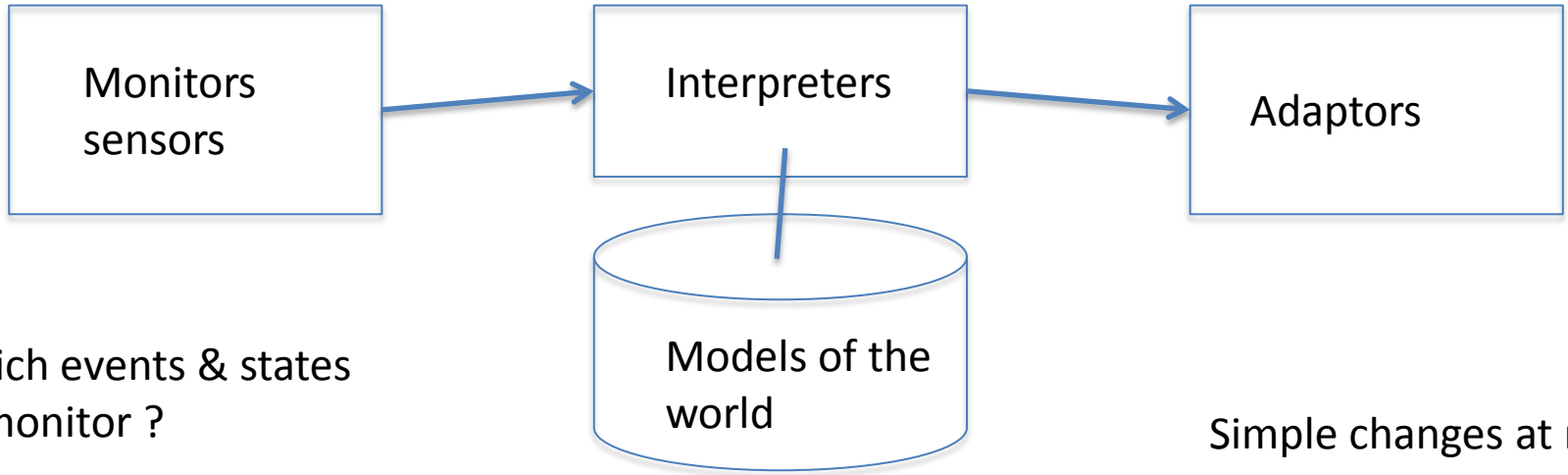
  (i) Deciding how to interpret the world

  (ii) Predicting Event – State interaction- difficult 20/20 foresight !

  but can take a systematic approach to scenario based exploration
  see Sutcliffe et al TSE 2000

  (iii) Predicting possible dangerous AR interactions- especially in moded systems

# Awareness Requs @ the event level




Which events & states to monitor ?

Active or passive sensors ?

What fidelity of monitoring ? (time, signal type..)

Interpreting simple Events

Event patterns

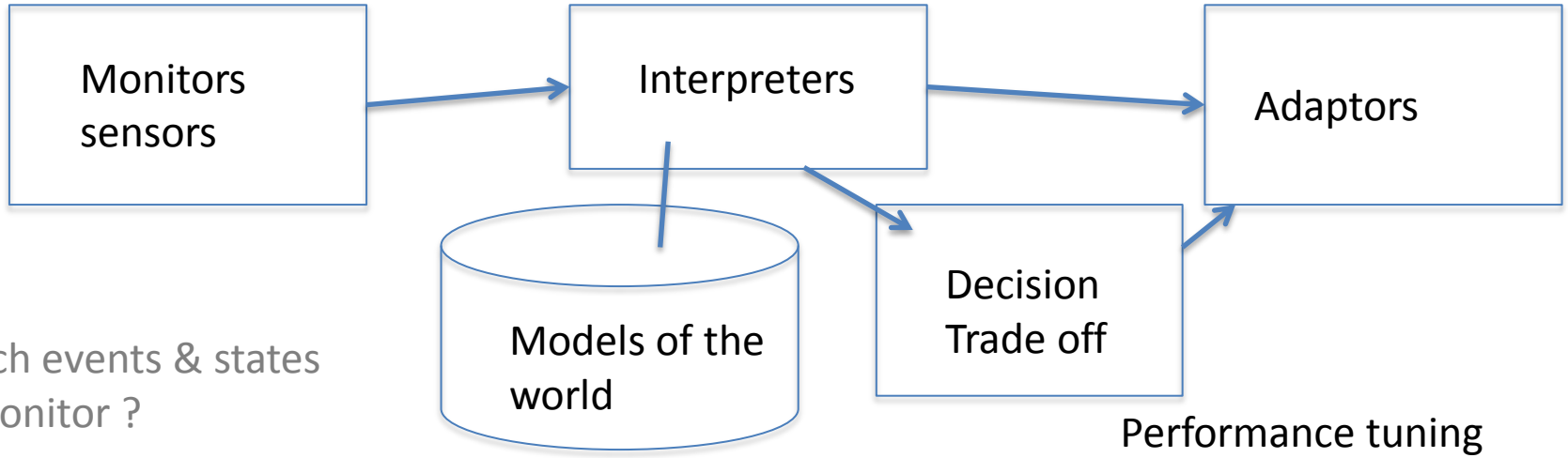
Higher order states

Simple changes at run Time

Rule/method level changes

Delegation

# Awareness Requs
# @ the Performance level




Which events & states to monitor ?

Active or passive sensors ?

What fidelity of monitoring ? (time, signal type..)

How long (time period)

Scope (population, area, etc)

Interpreting Event patterns

Higher order constructs states, intent, models

Data & Text Mining Learning Algorithms

Performance tuning

Component selection

Delegation

Requirements change {new designs, Versions, product life Feature adaptation}

# Conclusions

- Awareness requirements needs to distinguish between the Event & Performance levels
- ARs can be expressed as a generic architecture of the problem, plus types
- Models and taxonomies of generic monitors and adaptation strategies can guide analysis
- Methods for analysing ARs need to be developed driven from causal taxomonies {safety critical fault trees}

# Future work {projects}

- Develop taxonomy of monitors and specification method for ARs
- Method for specifying ARs in safety critical domains (inc state-event combination problem plus AR interactions)
- Develop method for specifying requirements for Interpreters in ARs w.r.t to the problem domain, also model based interpreters
- Adaptation strategies and trade off analysis, decision support and automated trade offs for adaptation

MANCHESTER 1824

The University of Manchester

# References


Hollnagel, E. (1998). *Cognitive Reliability and Error Analysis Method: CREAM*. Elsevier, Oxford.

Reason, J. (1990). *Human error*. Cambridge University Press, Cambridge.

Reason, J. (2000).Human error: models and management, . Cambridge University Press, Cambridge

Johnson,C.W. (2003) Failure in Safety-Critical Systems: A Handbook of Accident and Incident Reporting, University of Glasgow Press, Glasgow, Scotland, ISBN 0-85261-784-4.

Johnson C.W. and Holloway, C.M. (2003) A Survey of Causation in Mishap Logics, Reliability Engineering and Systems Safety journal, 80 (3), pp 271-291.

Levson N.C. (1995), Safeware, Safety systems and computers. Addison Wesley.

Oztekin, A., and J.T. Luxhøj, An Inductive Reasoning Approach for Building System Safety Risk Models of Aviation Accidents, Journal of Risk Research, Vol. 13, Nos. 3-4 (2010), pp. 479-499.

Sutcliffe, A. G. (1998). Scenario-based requirements analysis. *Requirements Engineering*, *3*(1), 48-65

Sutcliffe, A. G., & Gregoriades, A. (2007). Automating scenario analysis of human and systems reliability. *IEEE Transactions on System, Man and Cybernetics: Part A, 37*(2), 249-261.

Sutcliffe, A. G., Fickas, S., & Sohlberg, M. M. (2006). PC-RE: A method for personal and contextual requirements engineering with some experience. *Requirements Engineering*, *11*, 157-163.

Sutcliffe, A. G. (2003). Mapping the design space for socio-cognitive task design. In E. Hollnagel (Ed.), *Handbook of cognitive task design* (pp. 549-575). Mahwah NJ: Lawrence Erlbaum Associates..

Sutcliffe, A. G. (2002). *The Domain Theory: Patterns for knowledge and software reuse.* Lawrence Erlbaum Associates, Mahwah NJ.

Sutcliffe, A. G., Gault, B., & Maiden, N. A. M. (2005). ISRE: Immersive Scenario-based Requirements Engineering with virtual prototypes. *Requirements Engineering*, *10*(2), 95-111.

Sutcliffe, A. G., Maiden, N. A. M., Minocha, S., and Manuel, D. (1998). Supporting scenario-based requirements engineering. *IEEE Transactions on Software Engineering*, *24*(12), 1072-1088.

Sutcliffe, A. G., and Rugg, G. (1998). A taxonomy of error types for failure analysis and risk assessment. *International Journal of Human-Computer Interaction*, *10*(4), 381-405.

# Failure Causes & Monitors

| Source | Cause | Monitors |
|---|---|---|
| Organisations | Policies<br>Process<br>Cultures | Standards inspections<br>Observation<br>Performance tests |
| People | Capabilities<br>Skills, knowledge<br>Decisions | Behaviour monitors<br>Performance tests<br>Interviews |
| Hardware | Maintenance<br>Capacity, overload | Operating environment<br>Operational performance |
| Software | Bugs, specification errors<br>Performance | Operational performance<br>Event monitors |