EPICS
Engineering Proprioception in Computing Systems

An Outlook for Self-awareness in Computing Systems

Peter Lewis

University of Birmingham, UK

UNIVERSITY OF BIRMINGHAM

Overview of this talk

- Why might we be interested in self-awareness?
- Self-awareness: what is it anyway?
- Types of self-awareness.
- A framework for self-aware computing.
- Challenges.

Self-awareness?

Developments in computing systems

$\longrightarrow$

Large

ePICS
Engineering Proprioception
in Computing Systems

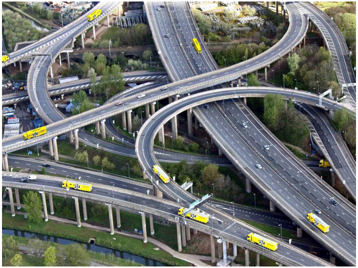
Developments in computing systems

$\longrightarrow$

Heterogeneous

Developments in computing systems

$\longrightarrow$



Uncertain

Developments in computing systems

$\longrightarrow$

Dynamic

Developments in computing systems

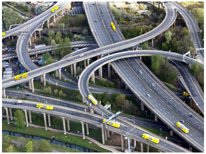
$\longrightarrow$

Decentralised

We're faced with systems that are...

Large

Heterogeneous



Uncertain

Dynamic

Decentralised

What do we want to do with these systems?

Requirements

typically describe measurable goals in terms of

- functionality,
- flexibility,
- performance,
- resource usage,
- costs,
- reliability,
- safety,
- security,
- ...

Behaviour is measured over multiple attributes.

Conflicts between goals.

Meeting requirements

How can systems "meet" these goals?

- Aim is achieve an acceptable trade-off between goals.
- Not a one-time optimisation problem!
- Deployment environments comprise dynamic and uncertain scenarios.
- In order to continue to manage trade-offs acceptably,
- We need: dynamic management of trade-offs at run-time.

Hypothesis:

- Systems which are **aware** of their own state, behaviour and performance can manage trade-offs between goals at run-time.
- This enables them to better meet their requirements in uncertain and dynamic environments.

Basic approach

We want to design systems as collections of **self-aware nodes**, which learn and interact to self-adapt and self-organise.

Key questions:

- (When) are there benefits associated with increased levels of self-awareness in computing systems?
- How can we design self-aware systems?

ePICS
Engineering Proprioception in Computing Systems

Self-awareness

Self-awareness

History

- Term first introduced around the start of the 20th century.
- Emerging field within Psychology during the 1960-80's.

Definition from psychology

- Morin (2006) defines self-awareness as
  "the capacity to become the object of one's own attention";
- *"The organism becomes aware that it is awake and actually experiencing specific mental events, emitting behaviours, and possessing unique characteristics."*

Self-awareness in psychology

Three key concepts:

- **Public** and **private** self-awareness.
- **Levels** of self-awareness.
- **Emergence** of self-awareness.

Public and private self-awareness: key concept 1

Two different ways in which a person can be self-aware:

- **Private self-awareness**:
  - **Knowledge of and based on phenomena internal to oneself.**
  - Internal (e.g. I am hungry, my battery level is low).
  - May include awareness of values, goals or behaviour.
- **Public self-awareness**:
  - **Knowledge of and based on phenomena external to oneself.**
  - Ones perspective on the environment.
  - E.g. Aware of relationship to others, effects of own behaviour, how others perceive me.
  - Subjective.

Duval & Wicklund (1972) / Goukens et al. (2007)

ePICS Engineering Proprioception in Computing Systems

Levels of self-awareness: key concept 2

1. **Ecological self** (basic stimulus-response, the absence of unconsciousness)
2. **Interpersonal self** (simple awareness of interactions with others, allows simple adaptive working)
3. **Extended self** (extension to reflect over time, aware of the existence of past and future interactions)
4. **Private self** (can process self-information: thoughts, feelings, intentions)
5. **Conceptual self** (abstract representation of oneself).

(Neisser, 1997)

Meta-self-awareness

The conceptual self also permits **meta-self-awareness**:
"being aware that one is self-aware" (Morin and Everett, 1990).

Emergence of self-awareness: key concept 3

In collective biological systems:

- The entire system can exhibit self-aware behaviour.
- Constituent parts may not exhibit self-awareness themselves.
- Global information is distributed, not present at a single point.

Examples:

- The brain,
- Ant colonies,
- Immune systems.

(Mitchell, 2005)

Emergence of self-awareness: key concept 3

Key points:

- A complex system may exhibit emergent behaviour *which appears globally self-aware*.
- No constituent part needs to possess information about the whole system.

Design implications:

- Need not require that a self-aware system possesses a global omniscient controller.
- Sufficient just to have knowledge of relevant parts.

Self-awareness in computing

We would like to take some of these ideas from psychology,

and translate and apply them to the design of computing systems.

$\longrightarrow$

A framework for self-aware computing

Why?

- Reduce confusion around the use of the term *self-awareness* in computing literature, by providing a common language and understanding.
- Better relate self-aware computing to self-awareness concepts in psychology; increase the pool of inspiration to draw on.
- Enable the principled engineering of self-aware systems, by identifying common ways to implement certain self-awareness capabilities, and evaluate and compare them.

Private and public self-awareness

Private self-awareness:

Based on knowledge of phenomena internal to the self.
E.g.

- Internal state,
- Individual goals and progress towards them,
- Own capabilities, available algorithms.

Public self-awareness:

Based on knowledge of phenomena external to the self.
E.g.

- Physical environment,
- Social environment (other individuals / nodes),
- Collective or others' goals and progress towards them.

Levels of computational self-awareness

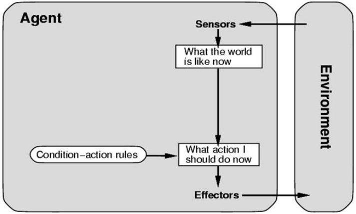
Computational framing of the **levels** of self-awareness:

- Ecological self $\rightarrow$ Stimulus-aware
- Interpersonal self $\rightarrow$ Interaction-aware
- Extended self $\rightarrow$ Time-aware
- Private self $\rightarrow$ Goal-aware
- Conceptual self $\rightarrow$ Meta-self-aware

Levels of computational self-awareness

Stimulus-aware:

- Knowledge of stimuli.
- Enables the ability to respond to events.
- Not able to identify reasons for stimuli.
- No knowledge of past/future stimuli.
- Can be either **private**, **public** or **both**.
- Examples: simple reflex agent, thermostat.



Levels of computational self-awareness

Interaction-aware:

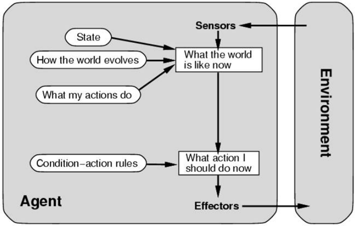
- Knowledge that stimuli and one's own actions form part of interactions with the social and physical environment.
- Able to distinguish between nodes and environments.
- Knowledge that actions can provoke, generate or cause specific reactions through feedback loops.
- Simple interaction-awareness may just enable reasoning about individual interactions.
- More advanced versions may involve knowledge of social structures such as communities and networks.
- This is a form of **public** self-awareness.
- Examples: bird alarm call, bittorrent client.

Bittorrent

Levels of computational self-awareness

Time-aware:

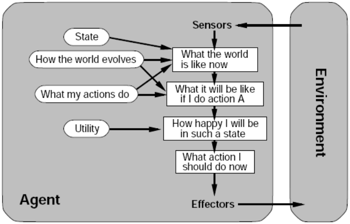
- Knowledge of historical and/or (likely) future phenomena.
- May involve explicit memory, time series modelling and/or anticipation.
- Can be either **private** or **public** or **both**.
- Examples: weather forecasting system, model-based reflex agent.



Levels of computational self-awareness

Goal-aware:

- Knowledge of current goals (e.g. goal states, utility etc.)
- May include knowledge of preferences and constraints.
- Permits acknowledgement of and adaptation to changes in goals.
- Can be either **private** or **public** or **both**.
- Examples: Satellite navigation system, utility-based and goal-based agents.



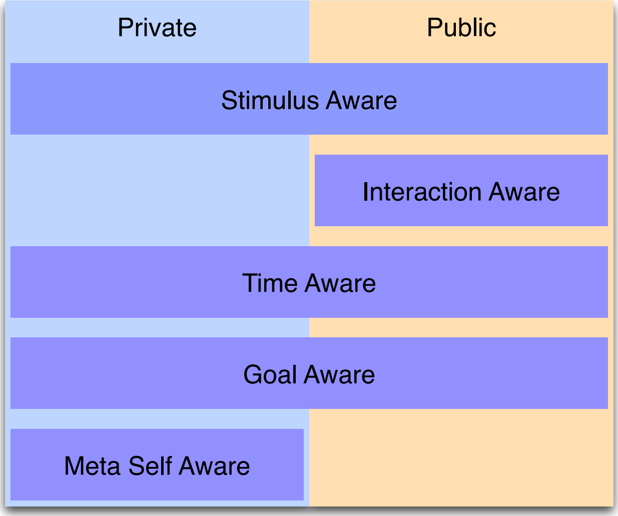
Levels of computational self-awareness

Meta-self-aware:

- Knowledge of the node's own self-awareness capabilities.
- May include knowledge of which level(s) of self-awareness are present, and how they are realised.
- Permits the ability to reason about the benefits and costs of maintaining certain levels of self-awareness, in terms of the node's goals.
- Enables online selection of a technique for realising other levels of self-awareness.
- This is a form of **private** self-awareness.
- Example: Mathematica's automatic algorithm selection, hyperheuristics.

Wolfram Mathematica 8

Illustration of self-awareness levels



Working definition of a self-aware computing node

To be self-aware a node should:

- Possess knowledge about its internal state *(private self-awareness)*.
- Possess knowledge about its environment *(public self-awareness)*.

Optionally, it might also:

- Possess knowledge of its interactions with the wider system (*interaction-awareness*).
- Possess knowledge of time, e.g. of past or likely future phenomena or effects (*time-awareness)*.
- Possess knowledge of its goals, e.g. objectives, preferences and constraints (*goal-awareness)*.
- Select what is and is not relevant knowledge *(meta-self-awareness)*.

Closed loops to cognitive robot collectives

Minimally, it is commonly considered that **self-aware systems** must simply operate based on a **closed feedback loop**.

Where self-aware systems differ is in what knowledge is available and collected, and how it is represented and used.

Perhaps it is less useful to ask *whether* a system is self-aware, but instead:

- What level(s) of self-awareness it possesses, and
- How its self-awareness capabilities are realised.

Implementing self-awareness capabilities

We've talked a lot about *what* self-awareness is,

but not a lot about *how* to do it.

- Different levels of self-awareness can be realised using a variety of techniques.
- Techniques can be used to realise several levels.
- Levels can be realised with different degrees of complexity.

Implementing self-awareness capabilities

Techniques can realise different levels:

- It depends on what the inputs are, and what the outputs represent.

Examples:

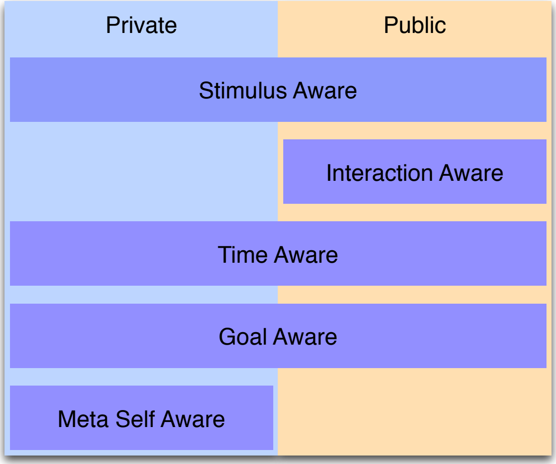
- Input could be current power usage -
  this might realise *stimulus awareness*.
- Input could be properties of other nodes -
  this might realise *interaction awareness*.
- Input could be performance of employed algorithms -
  this might realise *meta-self-awareness*.

Implementing self-awareness capabilities

Levels can be realised with different degrees of complexity:

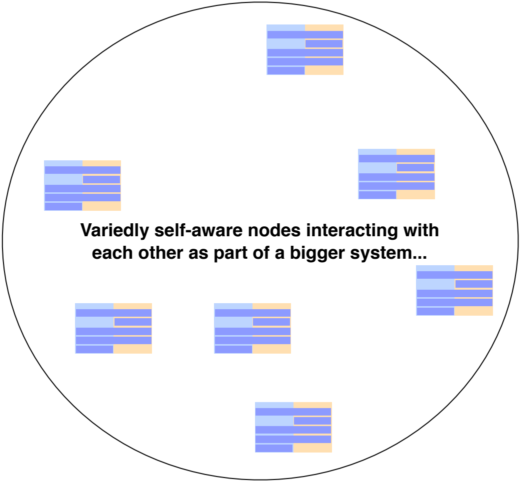
Level	Complexity $\rightarrow$
Stimulus	Threshold response $\rightarrow$ Linear response $\rightarrow$ Non-linear response
Interaction	Heuristic agents $\rightarrow$ Game theoretic $\rightarrow$ Social network
Time	MDPs $\rightarrow$ Use of memory $\rightarrow$ Anticipative agents
Goal	Objective function $\rightarrow$ Multi-attribute utility $\rightarrow$ Planning
Meta	Win-stay-lose-shift $\rightarrow$ Bandit-solvers $\rightarrow$ Ensembles

Nodes?

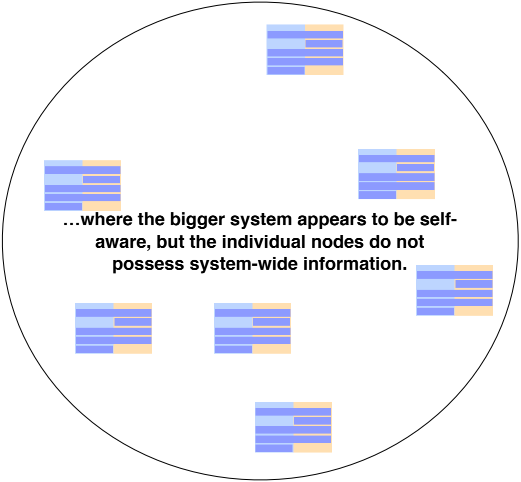


So where are we going to implement these things?

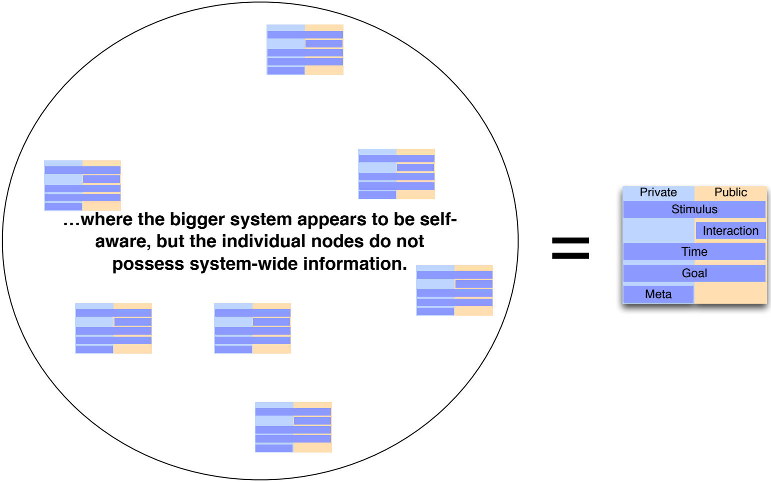
Collectives of nodes



Collectives of nodes



Emergent nodes



A node is a concept, not a box.

Challenge 1: meta-self-awareness

Typically:

	Scenario 1		Scenario 2		Scenario 3	
Algorithm	Mean	SD	Mean	SD	Mean	SD
1	4564.124	328.851	4408.050	86.483	8551.065	123.992
2	***5588.080***	***827.584***	3998.169	94.482	4038.210	1231.076
3	4829.110	417.563	4677.537	86.886	8247.195	142.299
4	5023.192	166.464	***4953.928***	***90.049***	6710.439	399.909
5	4995.957	107.555	4045.429	160.960	***9010.629***	***125.084***
6	4948.978	162.439	4901.786	95.389	8907.263	151.889

We identify the problem instance (scenario) and deploy the most appropriate algorithm.

Question:

Can self-aware systems do this for themselves?

Challenge 1: meta-self-awareness

Type 1 meta-self-awareness:

We have known scenarios and performance models, we want to know which technique to select.

- Experimental / theoretical results $\longrightarrow$ node's performance model.
- Task at the meta-self-aware level is to identify or **classify** the current scenario.
- Then the node selects the best algorithm for that scenario, according to the performance model.
- Challenge: identify the trade-off between the overhead of classifying the scenario and the relative benefit of selecting the "best" algorithm, and to make decisions based on this trade-off at run-time.
- Potential approaches: online classification, environmental change detection methods, ...

Challenge 1: meta-self-awareness

Type 2 meta-self-awareness:

We don't have an *a priori* known performance model, and must learn one online.

- Task at the meta-self-aware level is to learn the performance model, by actually deploying algorithms to perform the task, and monitoring their performance.
- Challenge: identify the trade-off between the overhead of building confidence in a performance model, and the relative benefit of selecting the "best" algorithm according to current knowledge.
- Potential approaches: online bandit solvers, ensembles, reinforcement learning generally, . . .

Challenge 1: meta-self-awareness

Type 3 meta-self-awareness:

We don't have an *a priori* known performance model, nor a set of pre-designed algorithms to choose from.

- Task at the meta-self-aware level is to build and test algorithms during run-time, deploying them and monitoring their performance online.
- Potential approaches: genetic programming, ...

Common key consideration:

A meta-self-aware system must manage the trade-off between possible future increased performance and the cost of learning to achieve this.

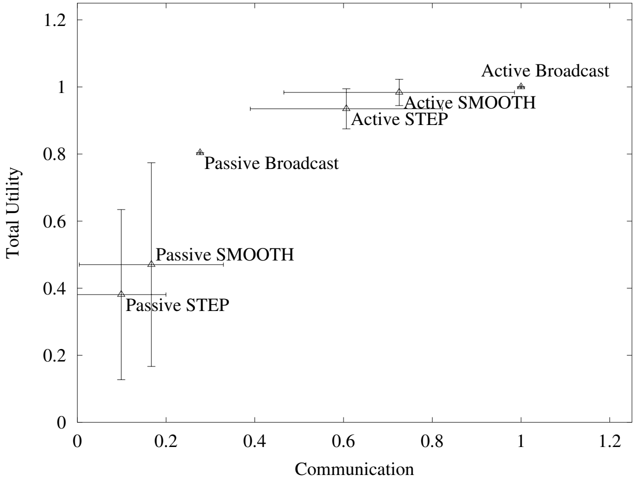
Challenge 2: claims about self-aware systems

Questions:

- How do we describe the benefits (or not) of computing systems being more self-aware?
- How can we expect them to behave? What claims can we make about these expectations?
- How should we describe the assumptions about scenarios and dynamics which we have considered?

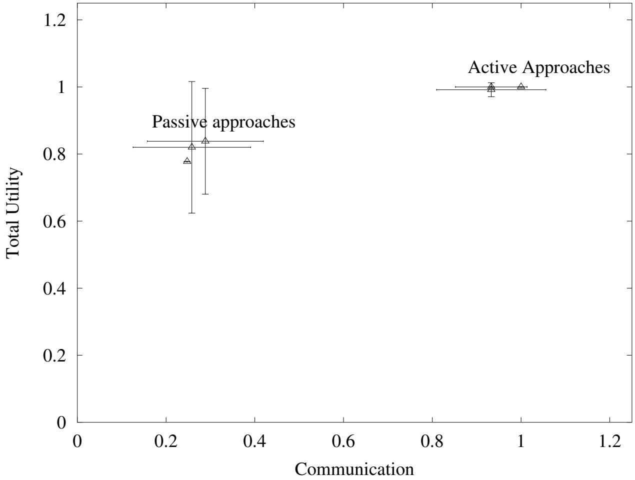
Challenge 2: claims about self-aware systems

Case Study: Handover in Distributed Smart Camera Networks



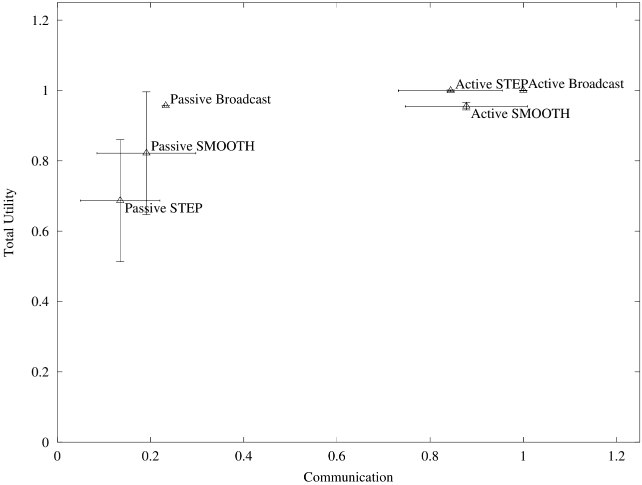
Challenge 2: claims about self-aware systems

Case Study: Handover in Distributed Smart Camera Networks



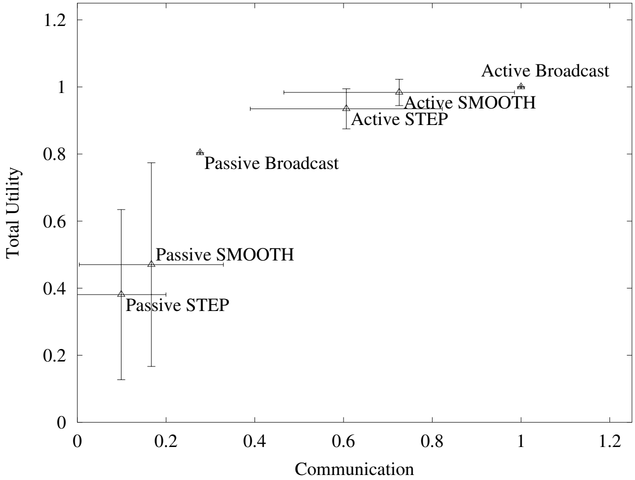
Challenge 2: claims about self-aware systems

Case Study: Handover in Distributed Smart Camera Networks



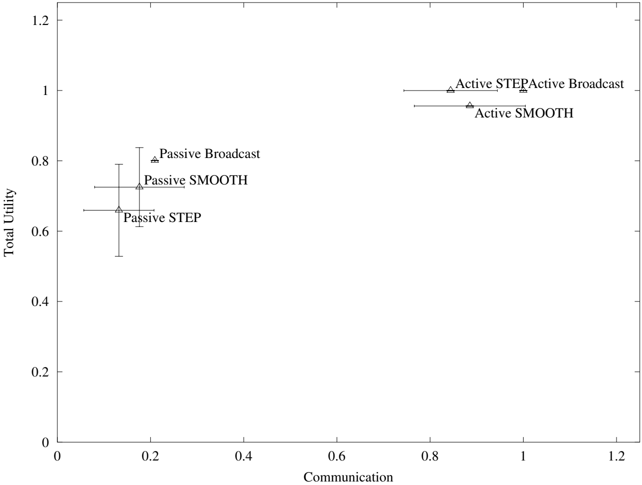
Challenge 2: claims about self-aware systems

Case Study: Handover in Distributed Smart Camera Networks



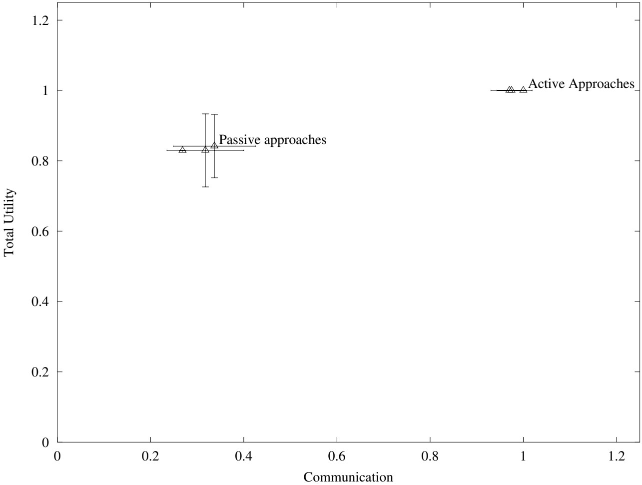
Challenge 2: claims about self-aware systems

Case Study: Handover in Distributed Smart Camera Networks



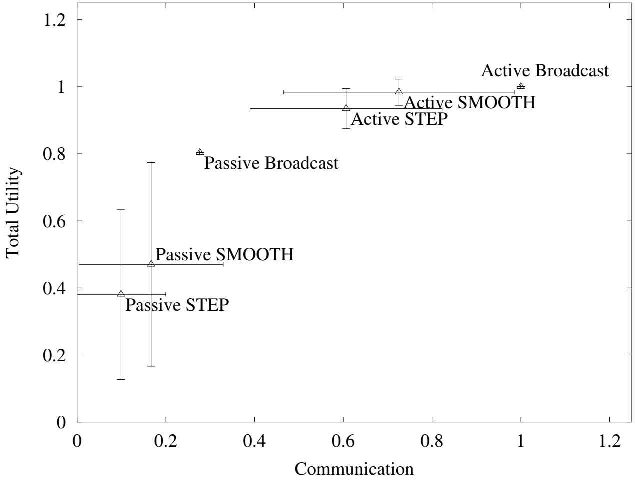
Challenge 2: claims about self-aware systems

Case Study: Handover in Distributed Smart Camera Networks



Challenge 2: claims about self-aware systems

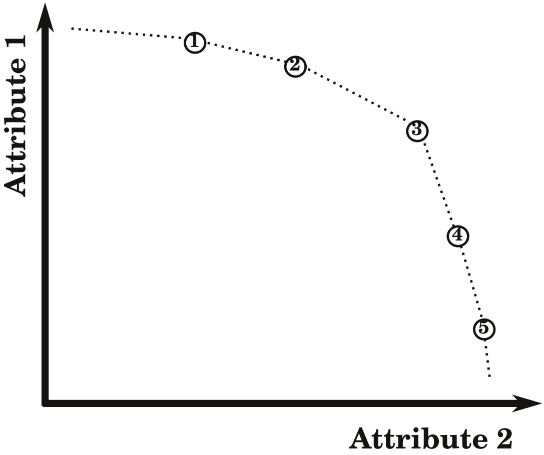
Case Study: Handover in Distributed Smart Camera Networks



ePICS
Engineering Proprioception in Computing Systems

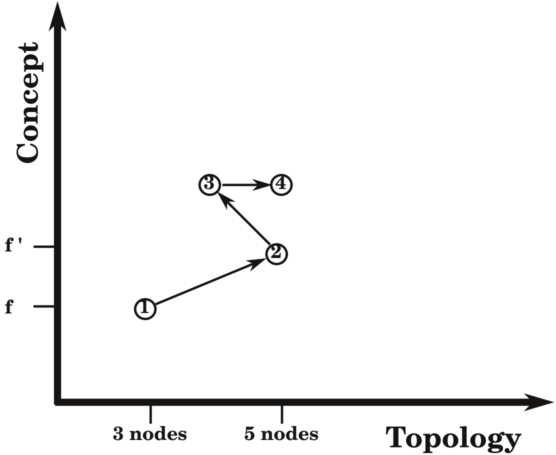
Challenge 2: claims about self-aware systems

A Single Problem Instance



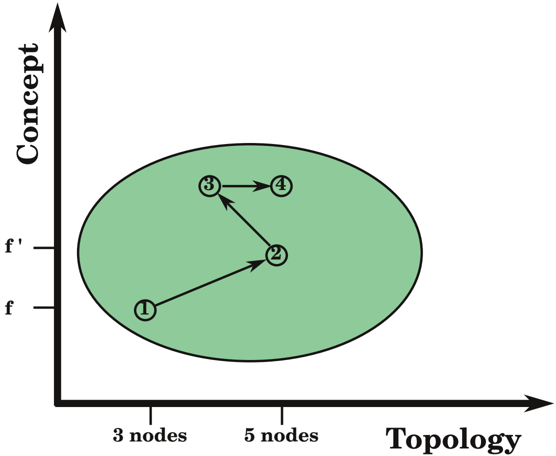
Challenge 2: claims about self-aware systems

Problem Instances at a Point in Time



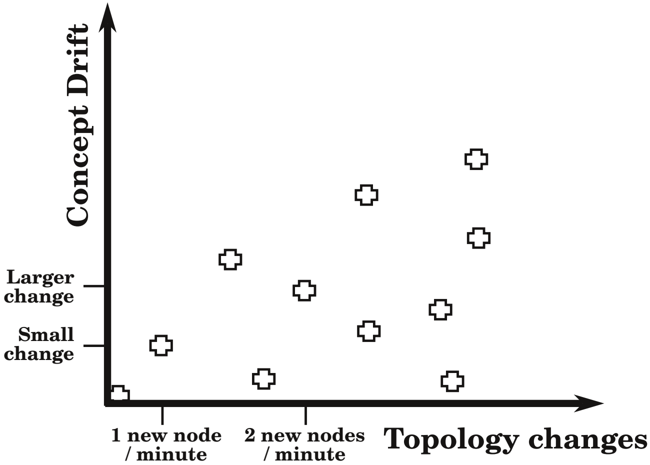
Challenge 2: claims about self-aware systems

The Space of Relevant Problem Instances



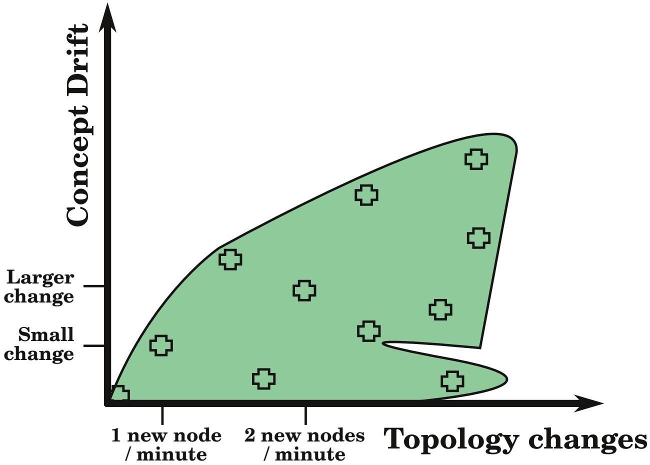
Challenge 2: claims about self-aware systems

Types of Dynamics



Challenge 2: claims about self-aware systems

The Space of Relevant Dynamics



Challenge 2: claims about self-aware systems

So. . .

In order to make claims about what self-aware systems will or will not do, we need to know what **the scope of the claims** is, with respect to:

- Problem instances,
- Dynamics,
- Goals,
- Preferences,
- . . .

Summary

Hypothesis:

- Systems which are **aware** of their own state, behaviour and performance can manage trade-offs between goals at run-time.
- This enables them to better meet their requirements in uncertain and dynamic environments.

Self-awareness framework based on three key concepts:

- Nodes can possess **public** and **private** self-awareness.
- The extent can be characterised by **levels** of self-awareness.
- Self-awareness can be an **emergent** capability; nodes are conceptual.

Summary

Our framework:

- Provides a common understanding of and language for self-aware computing systems.
- Provides a psychology-inspired basis for computational self-awareness.
- Aims to identify common ways to implement and evaluate self-awareness capabilities across the range of self-aware systems.

Key challenges:

- How do we implement meta-self-awareness capabilities, which are able to manage the trade-off between application goals and learning.
- How should we make claims about what to expect from self-aware systems operating in complex, uncertain and dynamic deployments?

More details...

P. R. Lewis, A. Chandra, S. Parsons, E. R. Robinson, K. Glette, R. Bahsoon, J. Torresen and X. Yao, ***A Survey of Self-Awareness and Its Application in Computing Systems***. In Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops (SASOW), pp 102-107. IEEE Press, 2011.

http://www.epics-project.eu/

http://www.prlewis.com/

Thanks for listening!

EPICS Engineering Proprioception in Computing Systems

References I

[1] S. Duval and R. A. Wicklund.
A theory of objective self awareness.
Academic Press, Oxford, UK, 1972.

[2] C. Goukens, S. Dewitte, and L. Warlop.
Me, myself, and my choices: The influence of private self-awareness on preference-behavior consistency.
Technical Report MO0702, Katholieke Universiteit Leuven, Belgium, 2007.

[3] A. Morin.
Levels of consciousness and self-awareness : A comparison and integration of various neurocognitive views.
Consiousness and Cognition, 15(2):358–71, 2006.

References II

[4] A. Morin and J. Everett.

Conscience de soi et langage interieur: Quelques speculations. [Self-awareness and inner speech: Some speculations].

Philosophiques, XVII(2):169–188, 1990.

[5] U. Neisser.

The roots of self-knowledge: Perceiving self, it, and thou.

Annals of the New York Academy of Sciences, 818:19–33, 1997.