THE IRISH SOFTWARE
ENGINEERING RESEARCH CENTRE

CLARITY
clarity-centre.org

sfi
science foundation ireland
fondúireacht eolaíochta éireann

Simon Dobson

UCD Systems Research Group
School of Computer Science
and Informatics
UCD Dublin IE

simon.dobson@ucd.ie
http://www.simondobson.org

Semantic challenges of adaptive systems

UCD DUBLIN

Overview

Autonomic and adaptive systems sometimes sound like old wine in a new bottle

- “Process algebra with a bit of control theory”

There are some fascinating new challenges for software engineering, especially in terms of analysis, design and correctness

My goal today

- Outline why there's value here
- Describe some of the approaches we're looking at, by way of a large environmental sensing project

Driving forces – 1

Complexity kills innovation and maintenance

- Systems become too hard to change

Many new systems have *very* challenging requirements

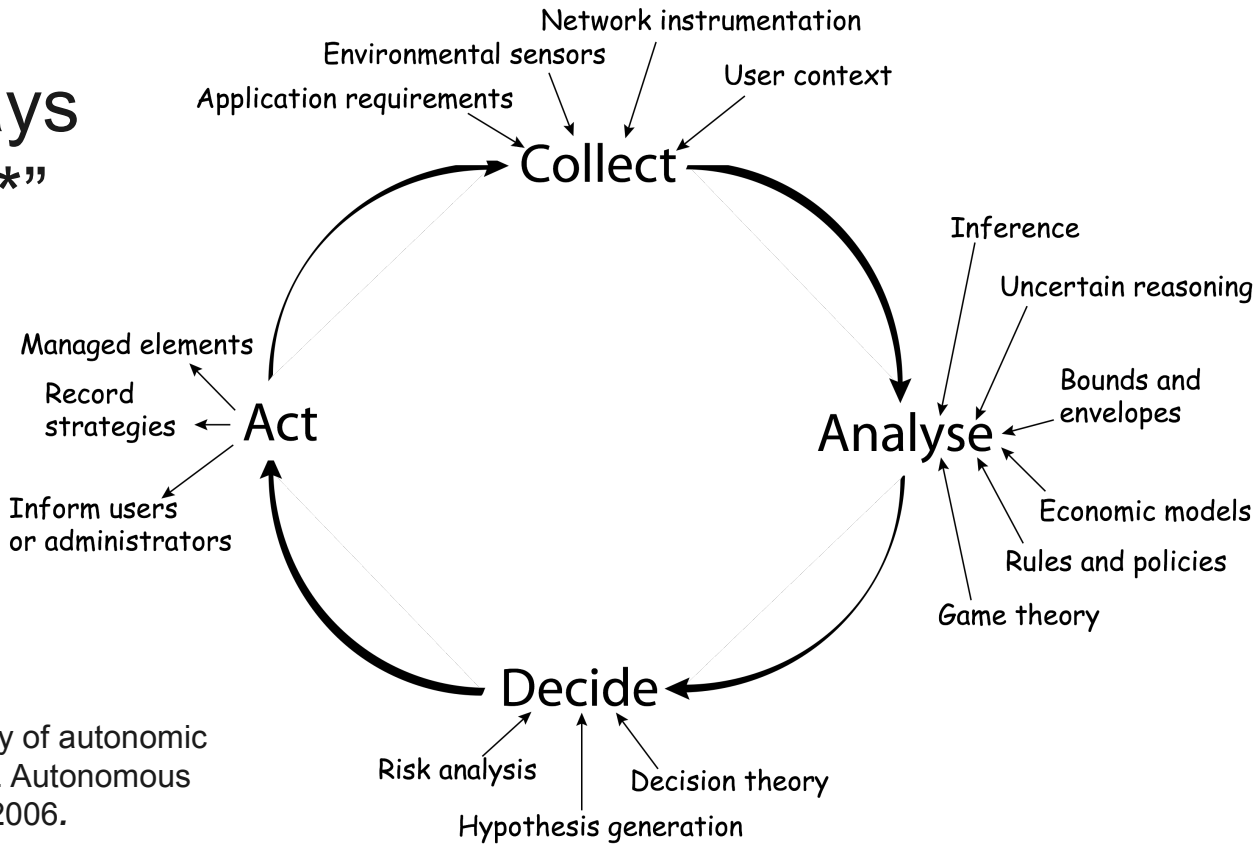
- Work without human intervention
- Adapt to maintain mission goals and profile
- Re-purposing and mission creep

- Data centres, environmental sensing, space exploration

Driving forces – 2

Autonomic systems

- Initially: decrease TCO through automation
- Recently: find well-founded ways of building “self-*” behaviours



From Dobson *et alia*. A survey of autonomic communications. ACM Trans. Autonomous and Adaptive Systems **1**(2). 2006.

Traditional approaches

We have a good ideas about how we describe the “correct” working of classical systems

There is a (single) “correct” behaviour – and our job is to find it and implement it

- Requirements, specification, validation, …

Quite a “closed universe”, in that the system, once defined, isn’t supposed to change

- Or at least such changes are handled off-line

What's new

In an “open universe” we want to be able to adapt behaviour in pursuit of mission goals

- Change algorithms, parameters, components
- ...without direct human intervention
- ...while being *guaranteed* to maintain the safety, liveness and other properties we need
- ...being able to trace high-level causes to low-level actions
- ...and without bringing the system down (much)

Central problem

The way in which we pull the knobs and levers, and the effect on the dials

The reason we have the system in the first place, defined externally to it

The dials, knobs and levers we can use to affect its operation

Given a system with a set of **high-level goals**, a particular set of **monitoring and management facilities**, and a changing environment, what is the **appropriate control strategy** that the system should take in **response to changes** in its environment in order to maintain, **as far as possible**, its goals?

This whole process is a dynamic, during the system's "live" operation

It may not *be* possible, and so we need a fallback strategy in case we fail

Abstractly – 1

We can view almost any system S as a mapping from inputs to outputs whilst maintaining an internal state

$$N_S : State_S \times Input_S \rightarrow State_S \times Output_S$$

What adaptivity does is to “lift” this “next state” function to include the external context

$$N_S : Context_S \times State_S \times Input_S \rightarrow State_S \times Output_S$$

And apply some control theory to make sure we don't diverge

However, it doesn’t do this arbitrarily

Abstractly – 2

We don't end up with just *any* function of three variables

- Some relationship between the original and the contextualised behaviour(s)
- There's a "shape" to the context that's "reflected" in the lifted, contextualised function

Dobson and Nixon. More principled design of pervasive computing systems, LNCS 3425. 2004.

So we need some way of capturing this "shape", and of *generating* the lifted function

- Make sure the repeated use of N_S leads to a system that preserves the properties we want

Which brings us to…

What is the correct semantic basis for *engineering* adaptive systems?

- Focus on adaptation, at the expense of the details of exactly what behaviour will be exhibited

  Often less interested in *what happens* than in how what happens *changes over time*

- Provide a clear mapping between causes (in the environment) and effects (in the system)

  Support analysis and design tasks, as an integral part of system functionality, *not* merely suring design, coding and commissioning

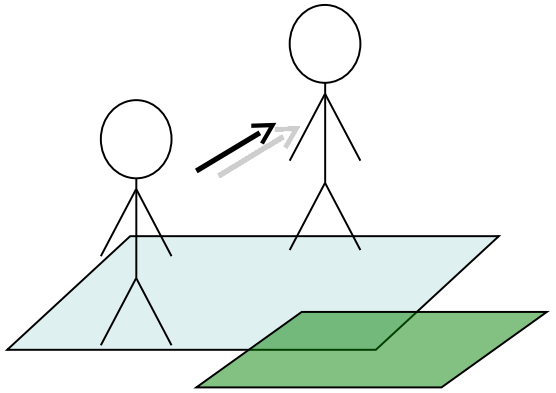
- Handle the uncertainty of sensor data and other information sources cleanly

  Don't over-react to changes that may simply happen because of low-resolution sensing or noise

Not all changes matter

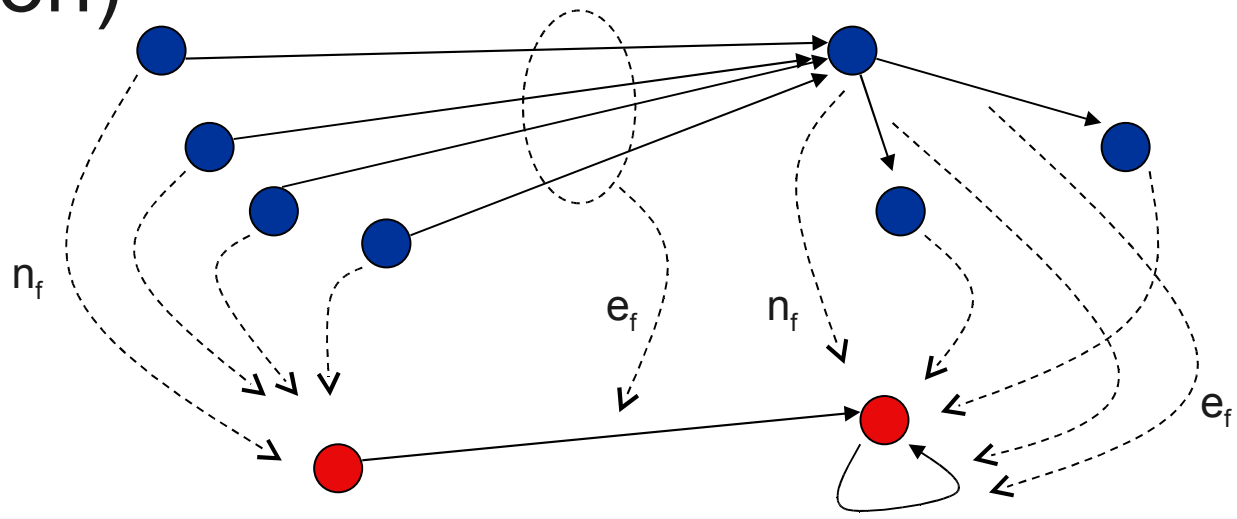
Simple location-awareness

- Change behaviour as user moves
- Not all movement matters, only "significant" crossings of boundaries



Induce a fibre structure between context and behaviour (situation)

Dobson and Ye. Using fibrations for situation identification. Proc. Workshop on Combining theory and system-building at PERVASIVE 2006.



Isolating “real” changes

Distinguish contextual changes that cause changes of behaviour from those that don't

- Edges within fibres = those transitions (observations, inference) that *do not change* the behaviour

Ye et alia. Using situation lattices in sensor analysis. Proc. IEEE PERCOM 2009..

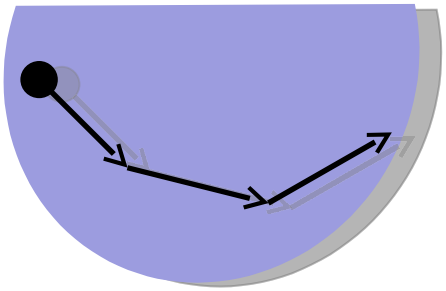
A better understanding of pervasive adaptation

- Fibre structures like this help to add rigour to the way in which we map context to situation

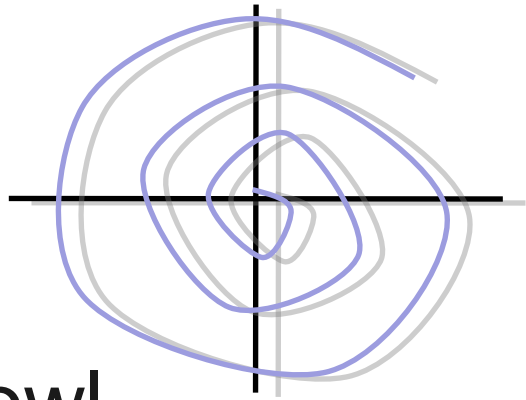
Doesn't capture the dynamics of adaptation as well as we'd like, but points in the direction of identifying “shape” in context

There's science…

Suppose we take a bowl and roll a ball around it

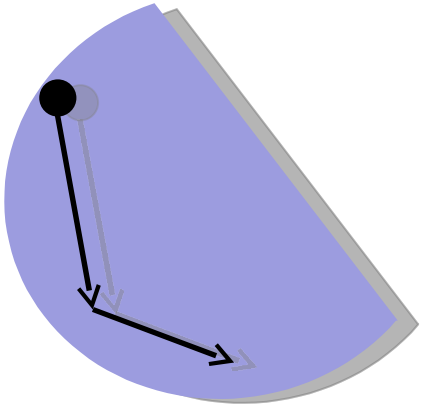


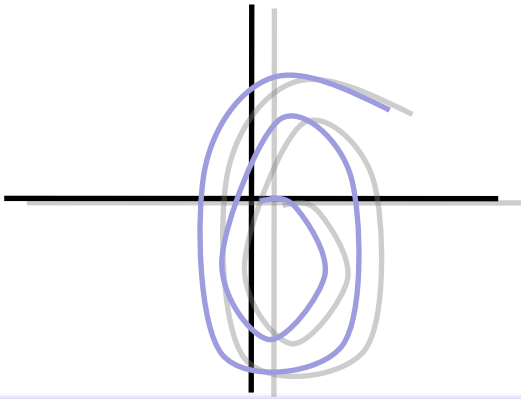
We can plot the ball's x-y position in the bowl and describe how it'll move, eventually coming to rest at the origin



If we change the orientation of the bowl and do the *same* action, we'll see a *different* dynamics

- But still determined by the initial conditions and the action





…and then there's engineering…

This situation is well-understood as a *dynamical system*, and can be applied to a wide range of situations

- Describe the *phase space* of the system – the valid tuples of all the variables – and the way one configuration changes into another

What we want is pretty much the opposite

- A way to *select* a particular behaviour given the constraints of an environment
- How the appropriate action changes as the environment changes

Neither of these problems is discussed, as far as we know, in the dynamical systems literature

Behavioural envelopes

What does it mean to be the phase space of an adaptive system?

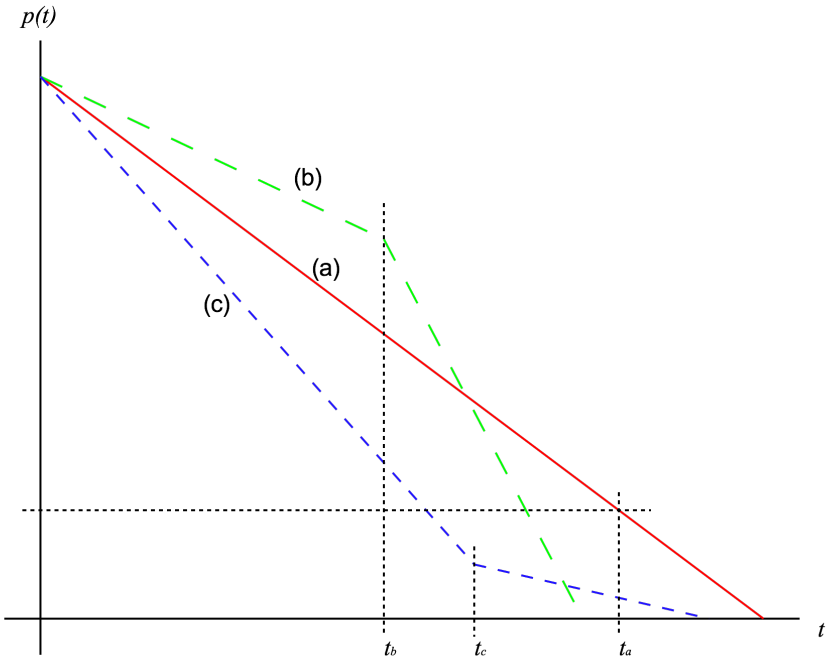
- Select all the variables in the system, identify all the legal tuples
- Define a *dynamics* within this sub-space: how the system will evolve from point to point
- Each control action changes the sub-space, or the dynamics, or both

In a physical system we *observe* the valid sub-space; in an engineered system we may want to *construct* it

A simple example

In a wireless sensor network we might see graphs like this relating power to time

- Lower use, and then a more precipitate drop
- A steep drop where we intervene at t_c to reduce load



From Dobson. An adaptive systems perspective on network calculus, with applications in autonomic control. Int. J. Autonomous and Adaptive Communications Systems **1**(3). 2008.

Changing the load changes the dynamics of the power behaviour

- Similar for data centre power management

UCD DUBLIN

Trading-off alternatives

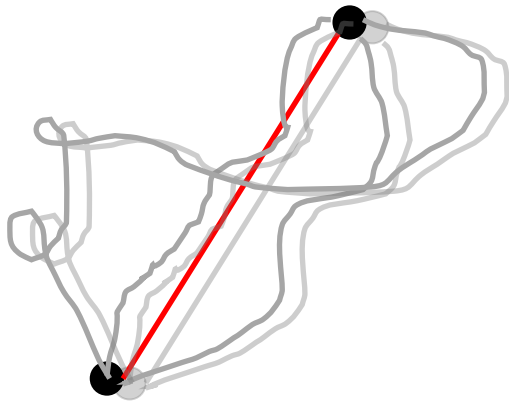
In many systems there may be several different *acceptable* dynamics we could pick

- In this previous example we might reduce communications *or* reduce sensing frequency *or* reduce accuracy – each of which saves power

The autonomic problem: decide *which* dynamics we want from the set of possibilities

Each path f_i between the two points is a function on $[0, 1]$, to which we can associate a path length $F(f_i)$

We then need to minimise $F(f_i)$ to find the shortest path



UCD DUBLIN

Case study: aquatic sensing

Much of Ireland's income comes from tourism and fishing, so we have a major interest in water quality

Much of the pollution comes from farming run-off (nitrates) from inland

How does the pollutant reach the sea? How does it disperse once it's there? What effects does it have on the sea and the coastline

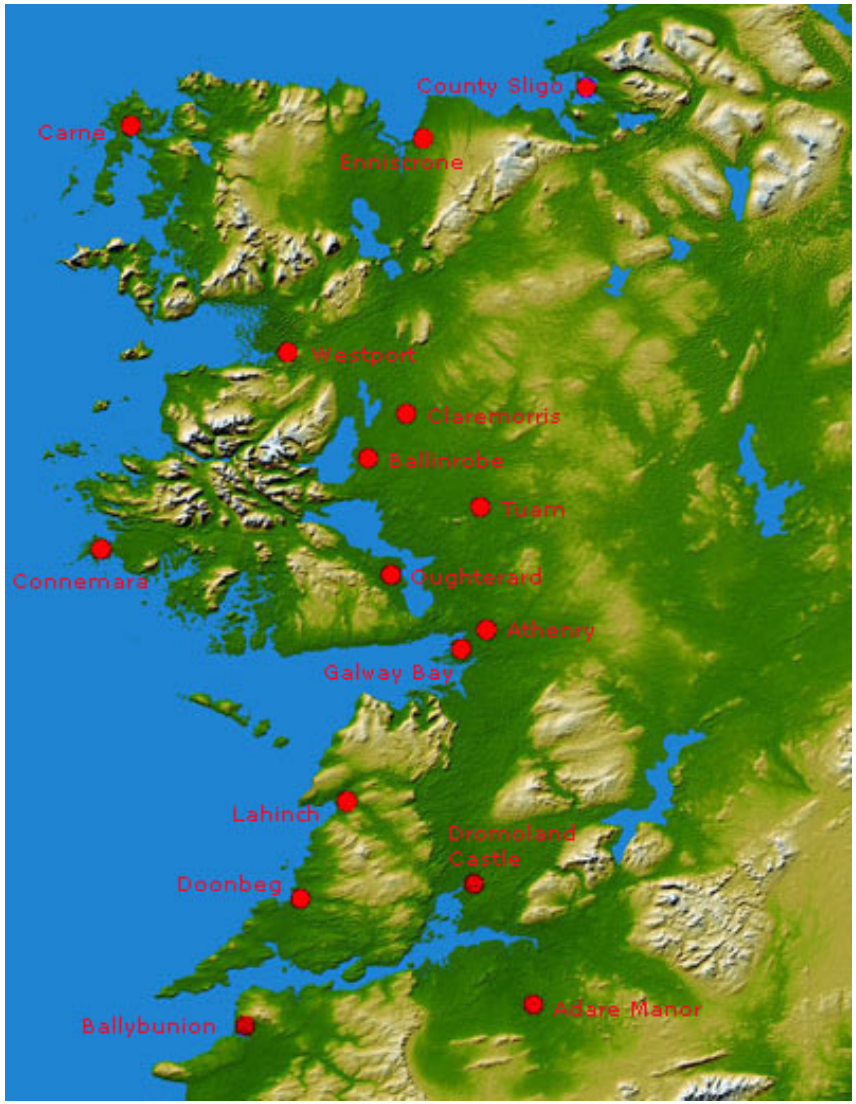
We – like every other country – *need* to know

The Irish aspect

Focusing research on areas off the west coast of Ireland

- Galway bay
- Shannon estuary

How can we mount an effective sensing mission in these busy areas?



UCD DUBLIN

Missions and mission goals – 1

Mission goals are almost always a trade-off

- ★ Provide high-resolution sensing of the area of interest
  - ...but also have a long life to get good value
  - ...and deal with partial failures in routing, sensing
- ★ ...and don't interfere with the environment being sensed
  - ...and did we mention the long life?

Clearly conflicts we have to resolve

Missions and mission goals – 2

In a lot of missions we can't make these trade-offs *a priori*

- Fixed sensing and communication periods (*duty cycle*) makes for predictable battery usage
- Too long a sensing period risks missing phenomena
- ...too short burns power sensing the uninteresting
- Too long a communications period risks losing data through failures, either local or remote
- ...too short runs down everyone's batteries

Adapting seems to make sense

Adaptive sensing – 1

We therefore want to *entangle* the management of a node with its sensing functions

- Make sensing a function of what's being sensed
- Increase frequencies when there's “something interesting going on”; reduce them otherwise

Makes things *much* more interesting

- Hard to model power and lifetime
- An additional factor to consider in terms of system correctness
- Better lifetime, performance, trust

Options

Network of static sensors

- Position in “interesting” places (or at random)
- In reality, constrained to stay out of the shipping lanes, scenic areas, fisheries, ...

Mobile sensors

- Move around, purposefully or at random
- *Try* to stay out of everyone's way, or be small enough to be run down without a problem
- Much harder control problem

Dealing with power

A typical mobile sensor requires power, both for its sensing/computing/communication and for its motion

Remove the latter by using yachts

- Wind power to move and recharge
- Indefinite lifetime
- Major planning problem in terms of how to move from *a* to *b* in given wind conditions
- Big enough for “real” sensors

UCD DUBLIN

Mission architecture

We envision a swarm of 1m (or larger) model yachts with sensor packages

- Why models? They're small, cheap, an already rigged for computer control by remote control
- Maintain communications either as a mesh or through some or all having longer-range radio

A collaboration between several large research centres and State agencies

- And chasing EU and NSF money...

UCD DUBLIN

CLARITY clarity-centre.org

lero THE IRISH SOFTWARE ENGINEERING RESEARCH CENTRE

sfi science foundation ireland fondúireacht eolaíochta éireann

Marine Institute Fiontar na Mara

Challenges

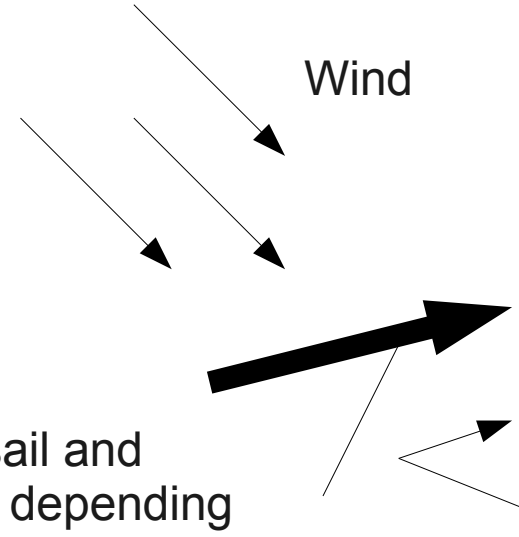
Too many to mention...

1. How can we sail a yacht under computer control so it goes where we want it to go?
2. How to we *decide* where we want to go?
3. How do we express this goal in a way we can analyse?
4. What is the best programming approach and/or language for highly sensorised adaptive systems?

For this talk we'll focus on the second and third

How to sail?

The “how” is horrifically complicated, but can be simplified



Anderson. The physics of sailing. Physics Today. Feb 2008.

A small number of sailing manoeuvres depending on direction of wind relative to desired direction

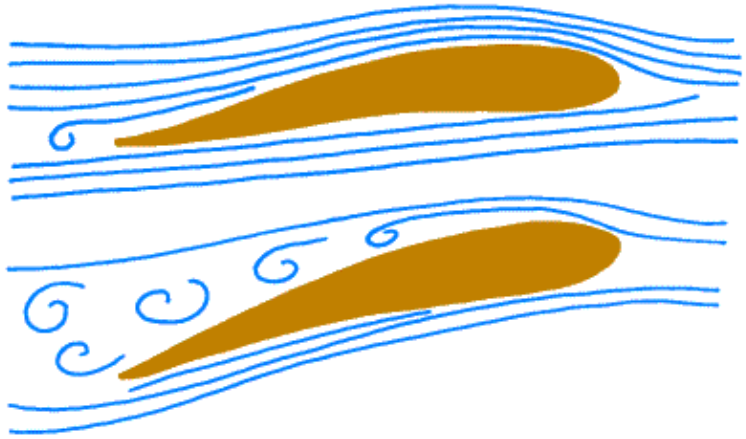
- A fairly classical AI planning problem

Where to sail?

Where would we want to sail to?

- Random direction – might find something interesting
- Static search pattern – can be tailored
- Dynamic pattern – need to know how to plan the pattern

Analogy: if you randomly sample an airflow over a wing, you'll get mostly laminar flow



UCD DUBLIN

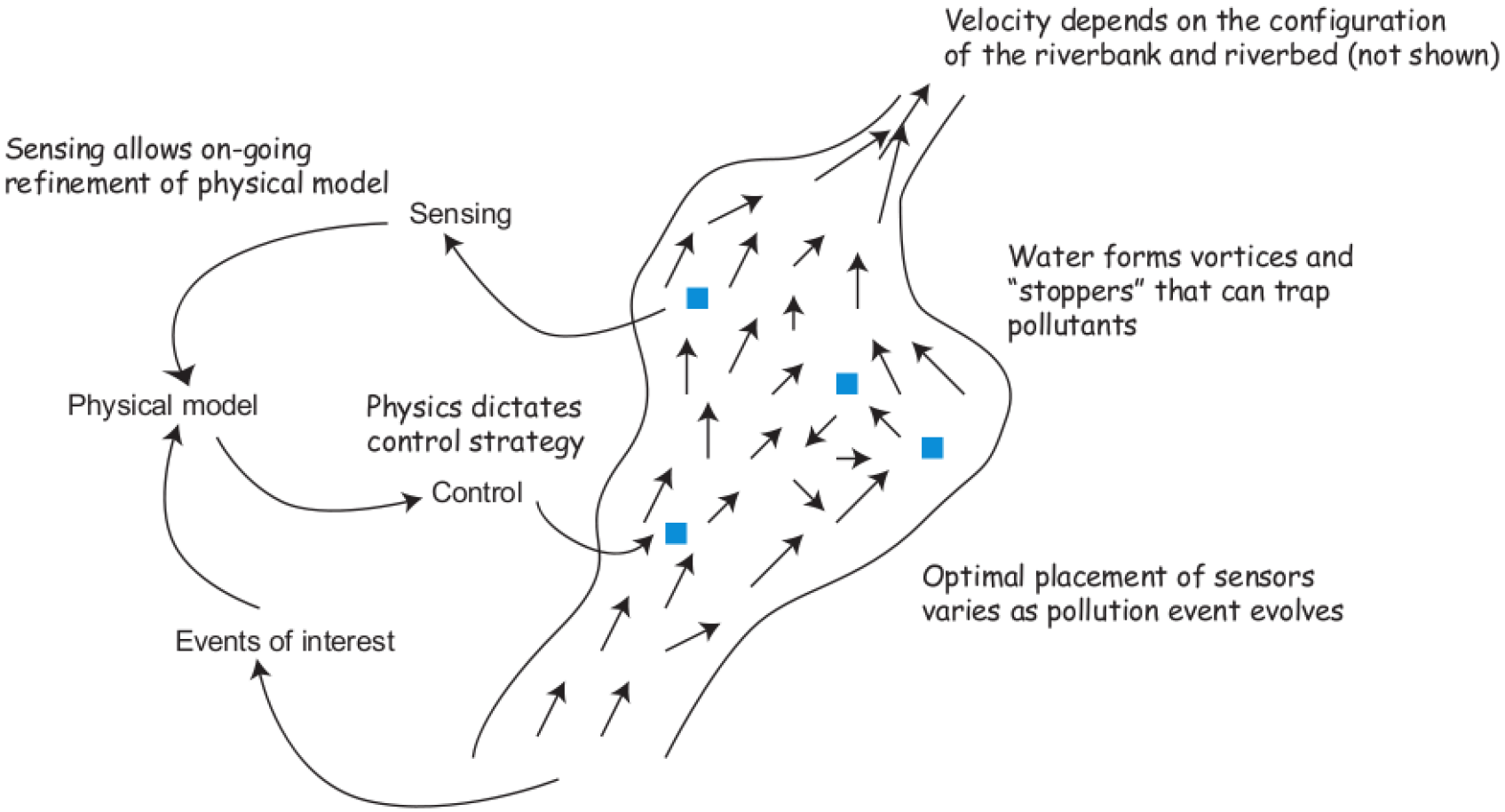
Knowing the physics

In order properly to plan a search pattern, we need to understand the physics of what we're searching for

- What constitutes an “interesting” place?
- How do these places evolve?

Although the detailed understanding of water flows is extremely complex, a naïve understanding will (perhaps) suffice for our purposes

A naïve understanding



From Dobson, Coyle, O'Hare and Hinchey. From physical models to well-founded control, Proceedings of IEEE EASe. 2009. To appear.

Controlling the swarm

We formulate this problem as one of maximising a value function for the swarm of yachts

Parameters

- The wind (vector)
- The flow field of the river (vector)
- The pollutant level at each point (scalar)
- The locations of the yachts (from GPS)

Value function

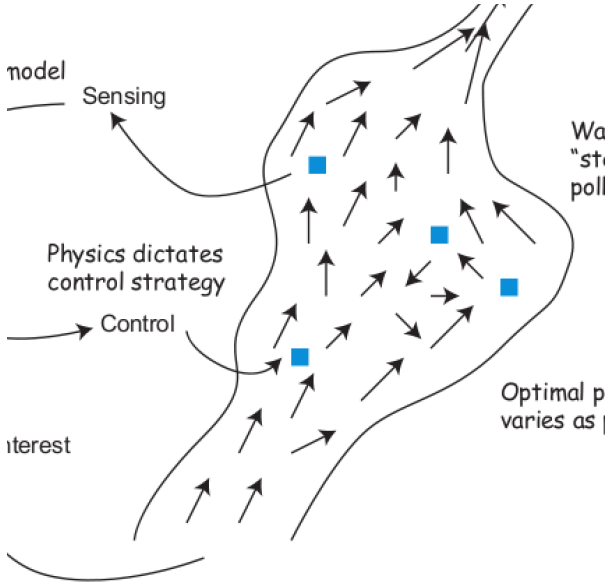
- Is each yacht sensing something interesting? Is the area being covered?

UCD DUBLIN

Defining the value function

We don't let have a really good definition

- Areas of high pollution
- Areas downstream of areas of high pollution
- Areas not being observed, to avoid missing other events



All seem to need a combination of local and global information, and a considerable amount of data exchange

Approach, once we have one

For any given scenario we can assign a value to any particular sensor constellation

- Non-unique values

For any particular sensor constellation we can define a movement that moves towards a better (or no worse) constellation

- Mostly a local operation, but requiring a global view of the scenario
- Must be balanced against what's possible, in terms of sailing against the wind etc

Cellai and Dobson. Generating a dynamic sensor coverage of an area. Work-in-progress for submission to ACM Trans. Sensor Networks.

UCD DUBLIN

Software engineering properties

It's important to realise that, although the model is defined globally, its implementation is neutral in terms of local and global decision-making

- May get better or worse results, and better or worse consumption of resources
- Provides a *semantics* against which to judge any solution, and against which to prove properties
- Decouple *specification* from *solution* to get better analysis – not common in the autonomics literature

Methodology

1. Obtain a physical model
2. Define capabilities of sensor nodes
3. For each model configuration define a “good” or “best” positioning constellation
4. Apply tactics to move current constellation towards a better (or no worse) one
5. Evaluate tactics by (for example) time to converge to best constellation – even though this will change in reality, and never be reached

Current state

- We can sail, in a straight line, downwind-ish, on our lake
- We can define simple models of fluid flow
- We have defined some value functions – none of which have been great
- We are starting to evaluate the combination of model and control integration
- We have a basic set of tactics for sailing

Three things to take away

Autonomic systems extend what we normally mean by system correctness, making it a dynamic process

Refining and re-deploying several mathematical techniques to entangle system models with those of environment, mission, etc

Lots of very timely, highly visible application areas to demonstrate well-founded results