The Emergence Engineers' Dilemma: it seems we can evolve emergence, or prove emergent properties, but not both

Alan FT Winfield

Bristol Robotics Laboratories

http://www.brl.ac.uk

EmergeNet4, York
19 April 2010

University of the West of England
BRISTOL

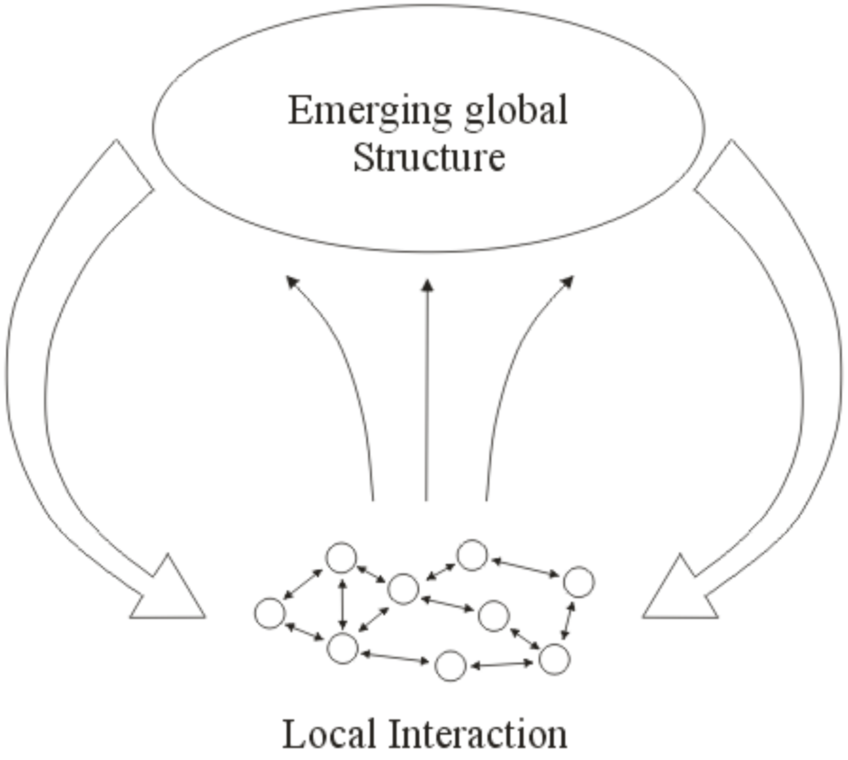
This Talk

- What is emergence and how can we engineer it..?
- Case study: Swarm Robotics
  - A definition for *Swarm Engineering*
  - An ad-hoc approach to swarm design
  - A principled approach to swarm design
  - How can we *prove* swarm properties?
  - What general conclusions can we (tentatively) draw from this case study?

What is Emergence?

- Emergent properties can appear when a number of simple elements interact in an environment, forming more complex global structures or properties

Emergent properties are not (easily) predicted or deduced from the behaviours of the single elements



From Lewton: Complexity - Life at the Edge of Chaos

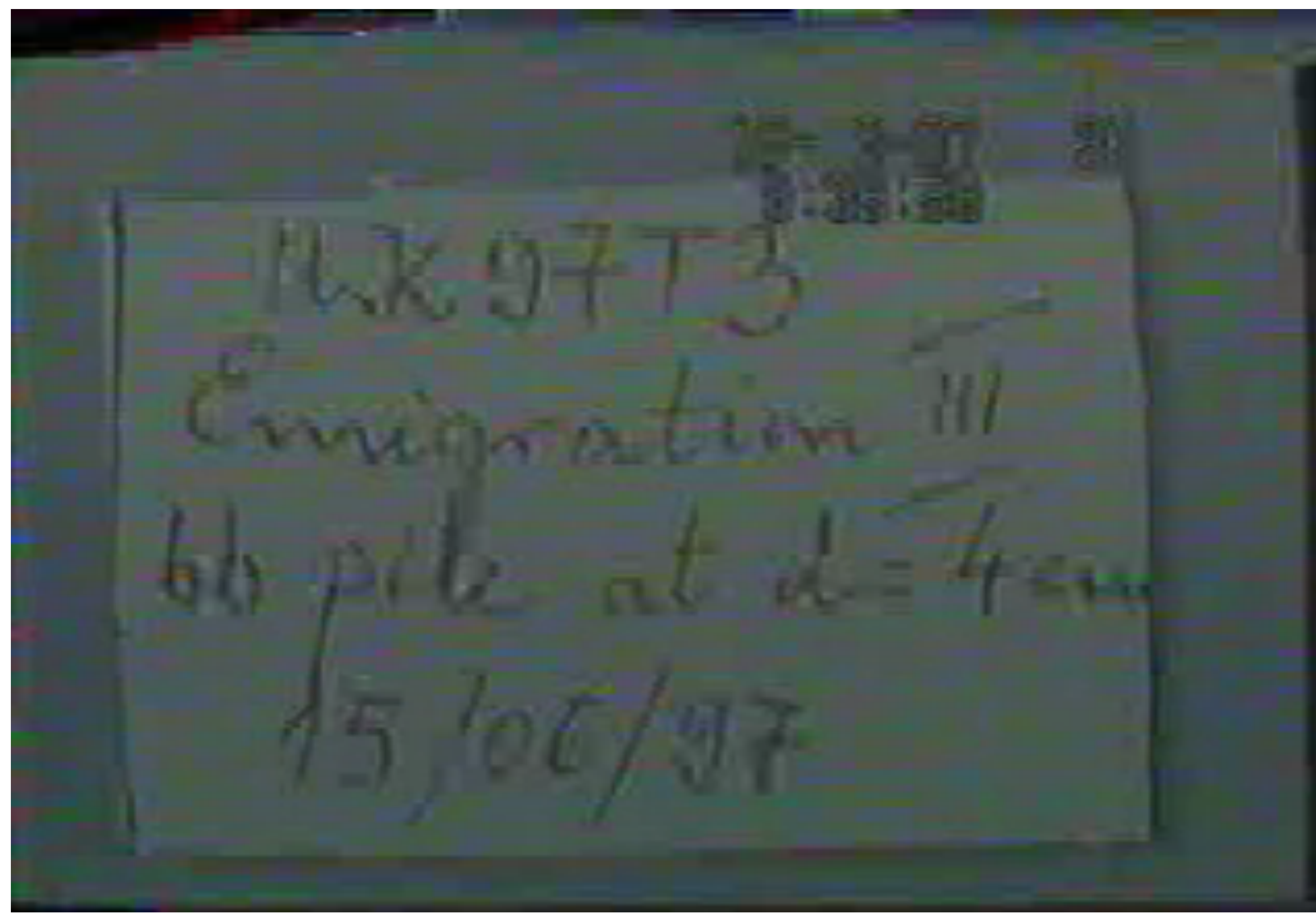
How can we engineer emergence?

- By designing the *simple elements*
  - i.e. designing the ways in which those elements interact with each other or with the environment
- By designing the *environment*
- Or, by designing *both* elements and environment
- But what are we designing for..?
  - *desired emergent properties*

Swarm Intelligence

– *“Any attempt to design algorithms or distributed problem-solving devices inspired by the collective behaviour of social insect colonies and other animal societies”* *Bonabeau, Dorigo and Theraulaz, 1999*

The ant ethologist tries to *reverse engineer* emergence



Leptothorax at work

Case study: Swarm Robotics is characterised by...

- Relatively simple, autonomous robots
- Fully distributed, de-centralised control
  - Exploitation of robot-robot and robot-environment interaction
  - Exploitation of explicit or implicit (stigmergic) communication
  - Self-organisation and emergence
- Scalability
- Robustness

But... can we engineer solutions with swarm intelligence..?

- What are the design principles involved?
  - how do we determine the *local rules* for each individual agent, in a principled way?
- How can we validate overall behaviours that are *emergent* properties?
  - notwithstanding these (difficult) questions...
- A powerful new engineering paradigm for large scale distributed systems..?

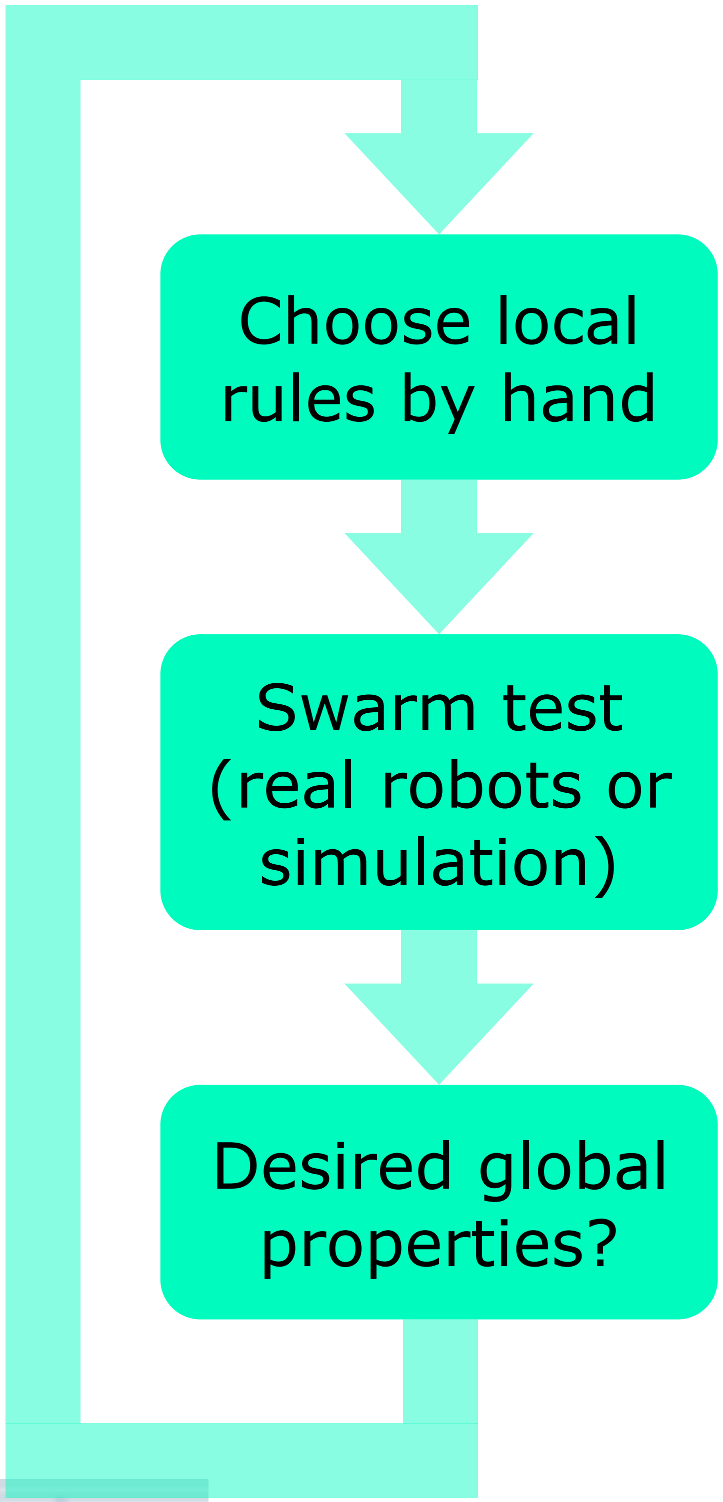
Swarm engineering

- The practice of *designing a complex system, using the Swarm Intelligence paradigm, which meets standards of analysis, design and test that would give sufficient confidence that the system could be employed in critical applications*

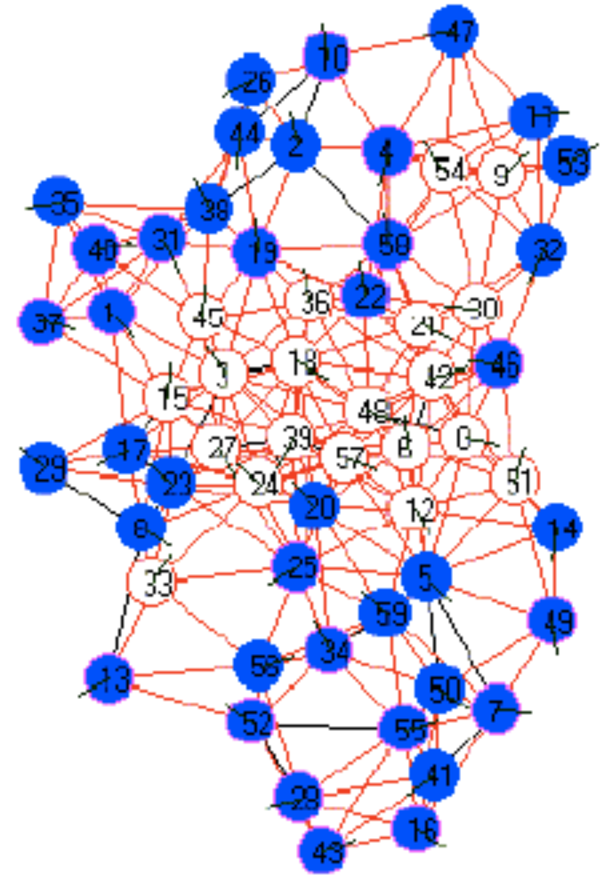
A. F. T. Winfield, C. J. Harper, and J. Nembrini. Towards dependable swarms and a new discipline of swarm engineering. In Erol Sahin and William Spears, editors, Swarm Robotics Workshop: State-of-the-art Survey, number 3342, pages 126–142, Berlin Heidelberg, 2005. Springer-Verlag.

Designing the local rules

swarm = superorganism



Ad-hoc
vs.
Principled approach

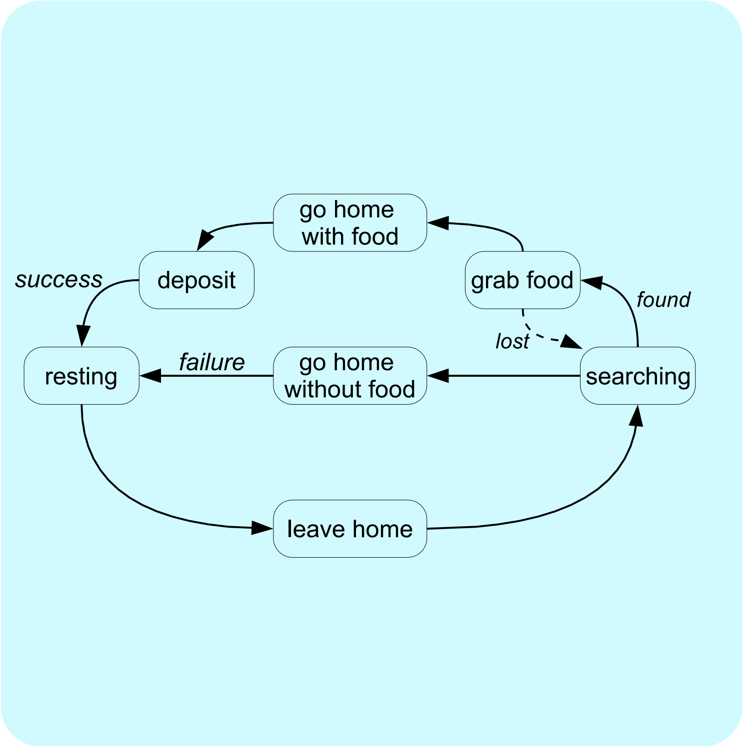


swarm = phenotype
global properties = fitness function
genotype determines local rules
Evolutionary swarm robotics

The robot controllers

Ad hoc approach

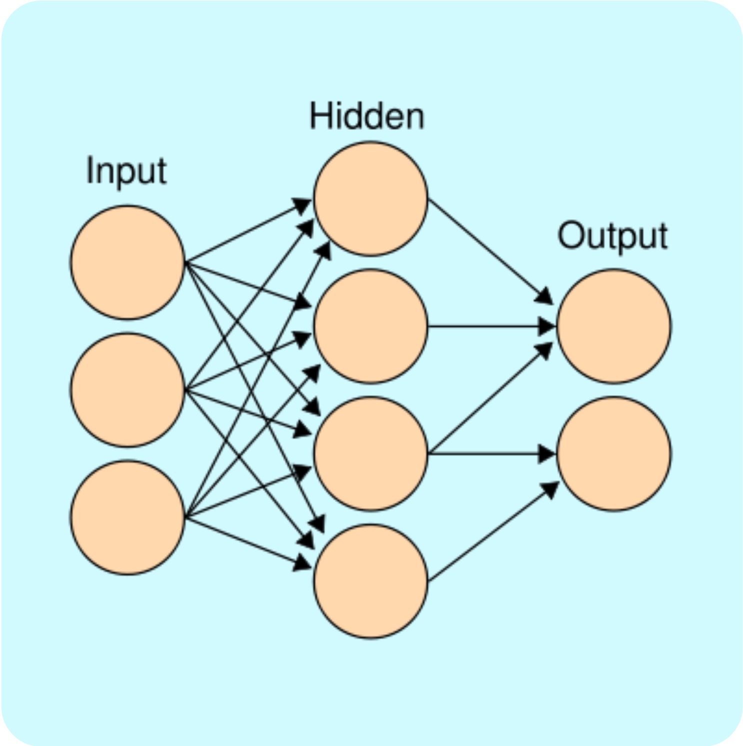
Sense inputs



Motor outputs

Evolved approach

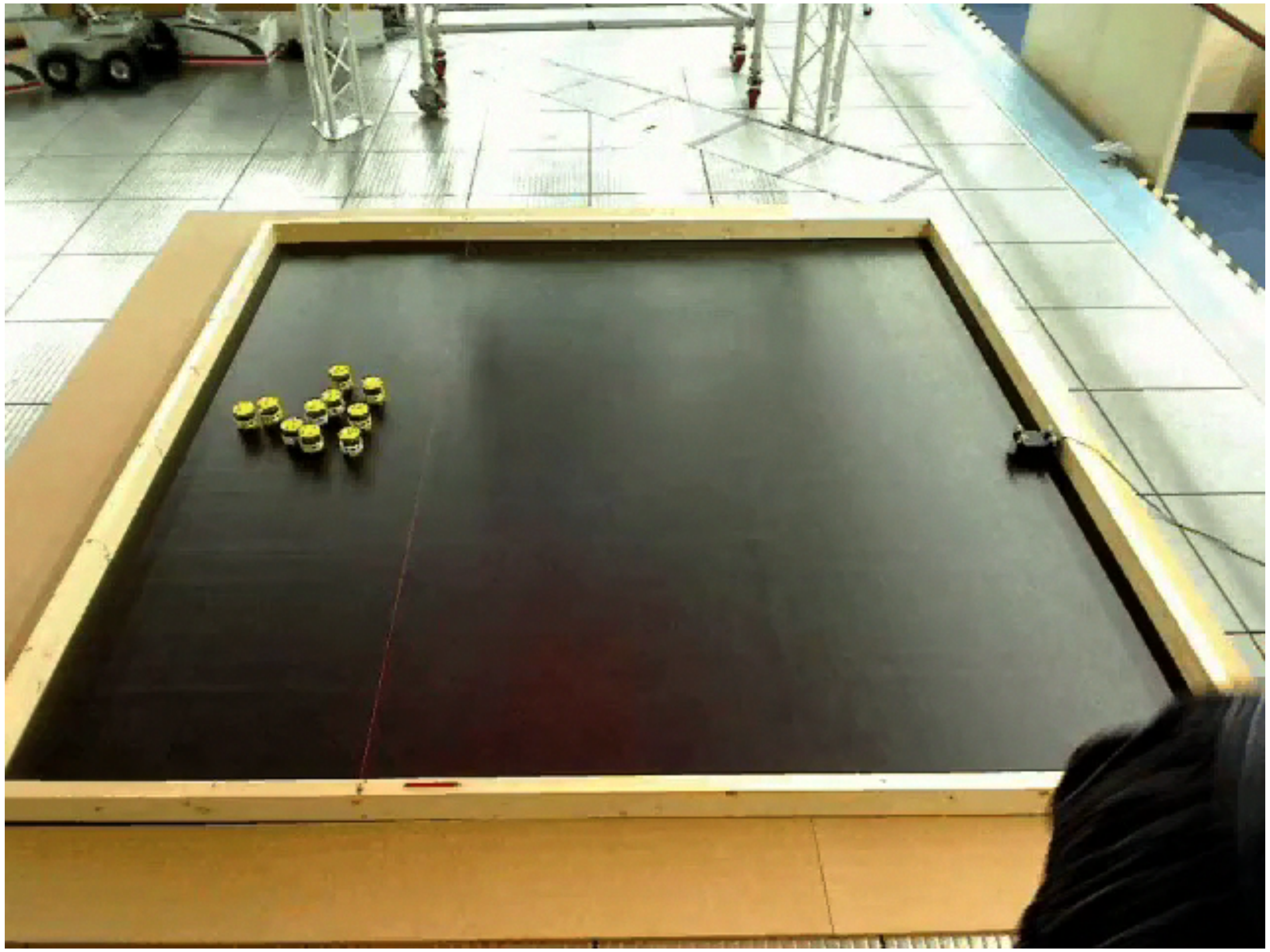
Sense inputs



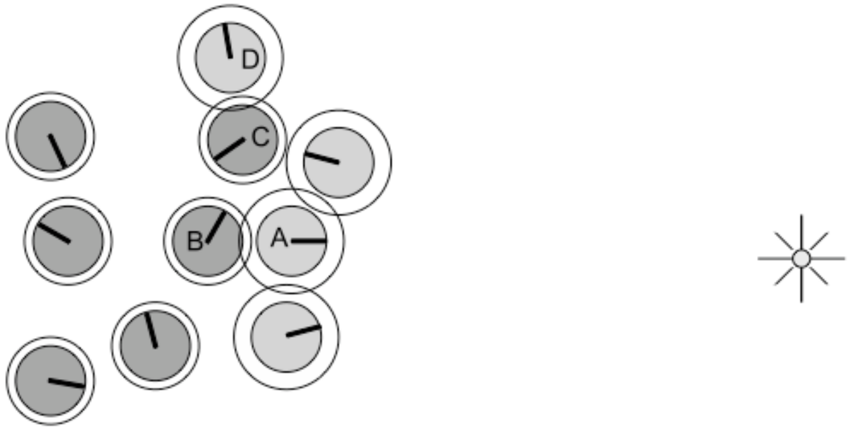
Motor outputs

But how can we validate the swarm?

Example: emergent flocking and swarm taxis

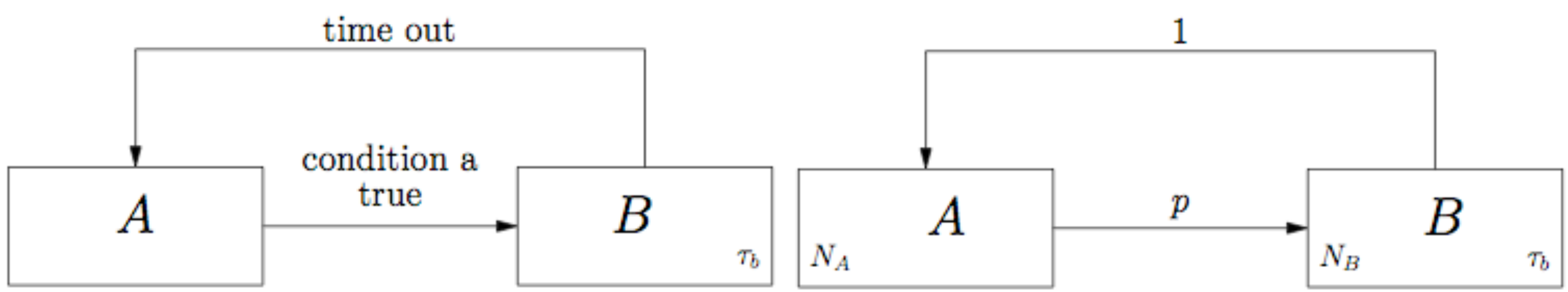


- How does it work...
- Robots have simple aggregation:
  - short range: obstacle avoidance (repulsion)
  - longer range: maintain number of connected neighbours (attraction)
- Each robot also has a simple beacon sensor
  - symmetry breaking mechanism: *illuminated* robots have a slightly larger avoid radius than *occluded* robots



Mathematical Modelling

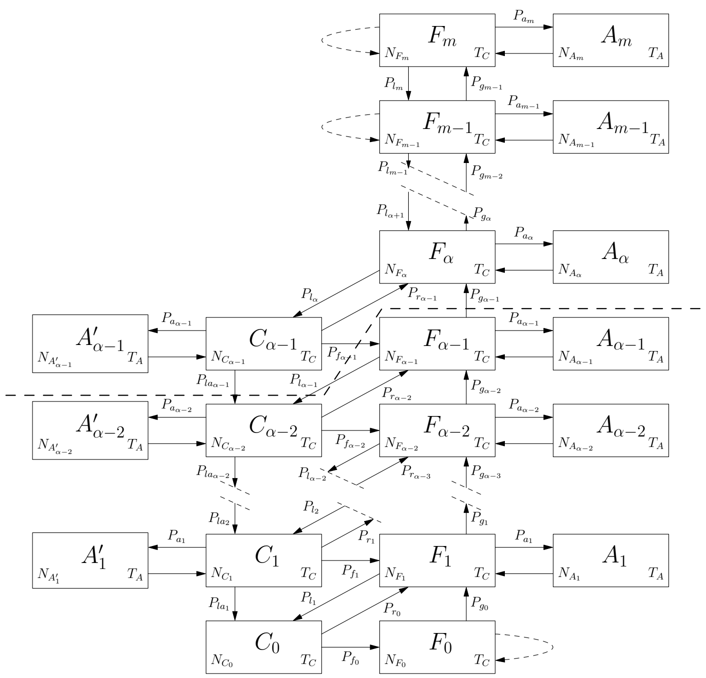
- We model the wireless connected swarm, by extending the probabilistic approach of Martinoli *et al**.
- We take the Finite State Machine (FSM)
  - express as an ensemble of probabilistic FSMs...which lead to a set of difference equations
  - geometrically estimate the transition probabilities
  - compare the model with experimental data

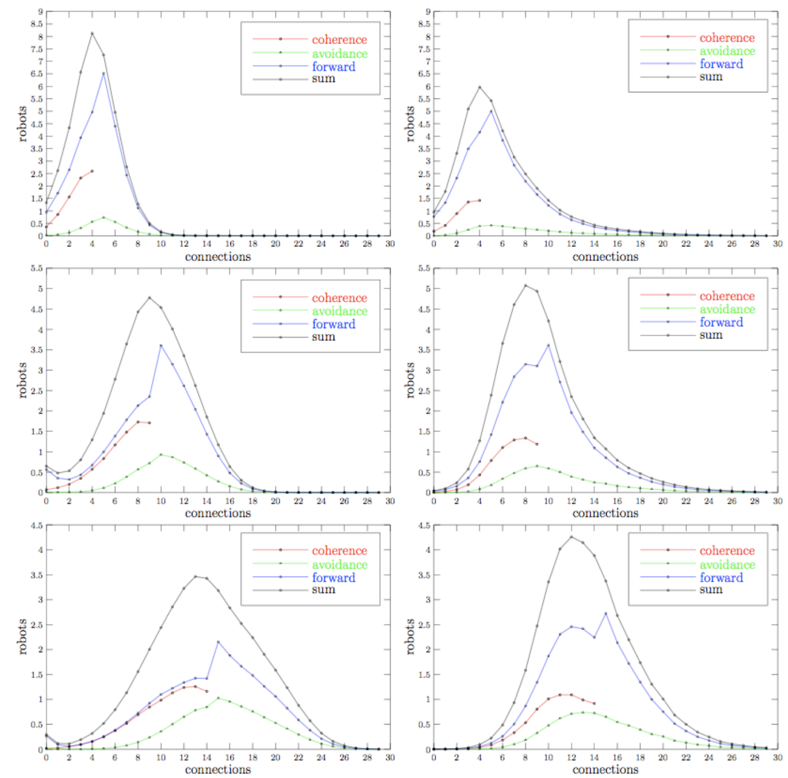


(a) finite state machine (b) PFSM

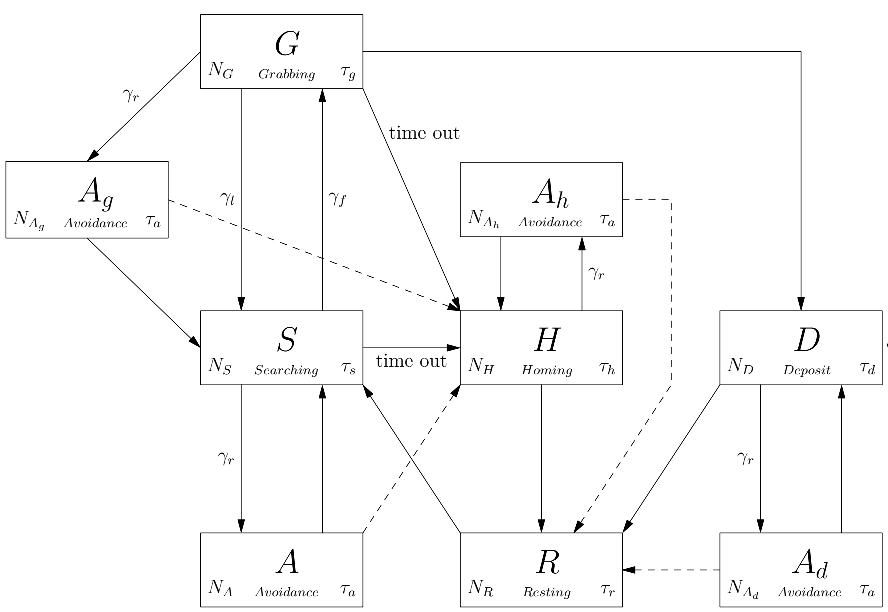
*See e.g. Martinoli, Easton and Agassounon, IJRR 23(4), 2004

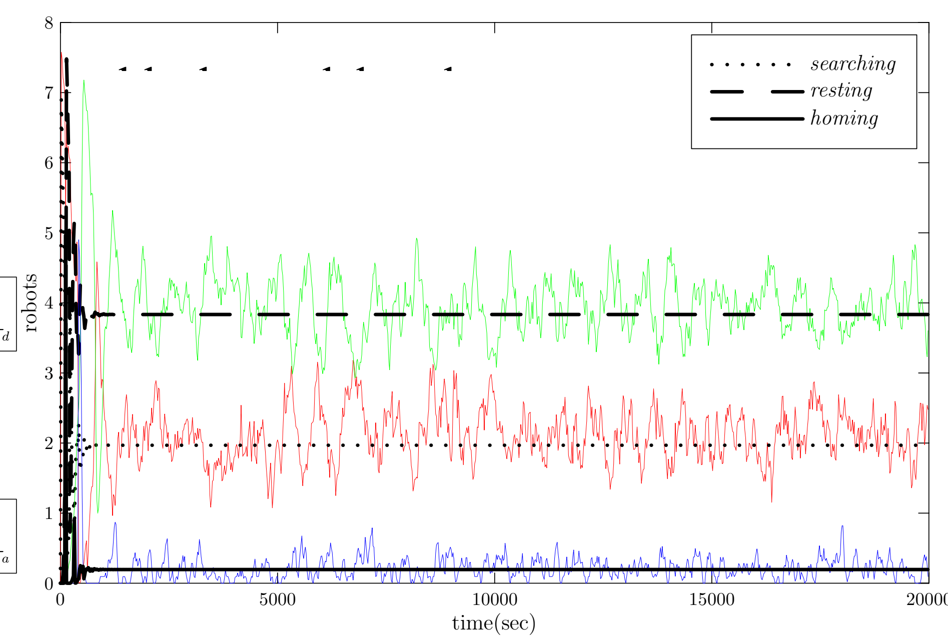
Models for flocking and adaptive foraging





Winfield AFT, Liu W, Nembrini J and Martinoli A, 'Modelling a Wireless Connected Swarm of Mobile Robots', Swarm Intelligence, 2 (2-4), 241-266, 2008.





Liu W, Winfield AFT and Sa J, 'Modelling Swarm Robotic Systems: A Case Study in Collective Foraging', Proc. Towards Autonomous Robotic Systems (TAROS 2007), pp 25-32, Aberystwyth, 3-5 September 2007.

Using Temporal Logic to Specify Emergent Behaviours

- We have investigated the use of a Linear Time Temporal Logic to specify (and possibly prove) emergent properties
- NASA have explored formal methods within the Autonomous Nano-Technology (ANTS) project
    - (Rouff et al, 2004)
    - however that work did not investigate a *temporal logic*

Some definitions

- Safety
  - the safety property specifies the set of *allowed* actions
- Liveness
  - the liveness property specifies the *dynamic* behaviour, that is, *all* possible eventualities
- Dependability requires that we establish *both* safety and liveness

A linear time Temporal Logic

- Extends classical logic with temporal operators,
  - $\bigcirc\varphi$ is satisfied if φ is true in the *next* moment in time
  - $\Diamond\varphi$ is satisfied if φ is true at *some* future moment in time
  - $\Box\varphi$ is satisfied if φ is true at *all* future moments in time
- Concurrency modelled by interleaving

Specify primitive robot behaviours

Specify the movement primitives, bottom-up

$$moveN(i) := (\bigcirc x_i = x_i) \wedge (\bigcirc y_i = y_i + a)$$

$$\begin{aligned} turn180Move(i) :=& \\ &(\theta_i = S) \wedge (\bigcirc\theta_i = N) \wedge moveN(i) \vee \\ &(\theta_i = W) \wedge (\bigcirc\theta_i = E) \wedge moveE(i) \vee \\ &(\theta_i = N) \wedge (\bigcirc\theta_i = S) \wedge moveS(i) \vee \\ &(\theta_i = E) \wedge (\bigcirc\theta_i = W) \wedge moveW(i) \end{aligned}$$

One of the four possible state/movement transitions

$$\begin{aligned} forwardNotConnected(i) :=& \\ &(motion_i = forward) \wedge \neg connected(i) \wedge \\ &(\bigcirc motion_i = coherent) \wedge turn180Move(i) \end{aligned}$$

Safety

Valid component (robot) actions

$$CompAction_i := forwardConnected(i) \lor forwardNotConnected(i) \lor coherentNotConnected(i) \lor coherentConnected(i)$$

$$Safety_i := \pi_i \land CompAction_i \land \forall j \in robotSet \setminus \{i\}.idle(j) \lor \neg\pi_i \land EnvAction_i \land idle(i)$$

Valid environment (other robot) actions

This specification ensures that only 1 robot is taking action at a time (interleaving)

Liveness

$$Liveness_i :=$$

$$(\pi_i \wedge (motion_i = forward) \wedge connected(i) \Rightarrow moveF(i))$$

$$\wedge$$

$$(\pi_i \wedge (motion_i = forward) \wedge \neg connected(i) \Rightarrow (turn180Move(i) \wedge \bigcirc motion_i = coherent))$$

$$\wedge$$

$$(\pi_i \wedge (motion_i = coherent) \wedge \neg connected(i) \Rightarrow moveF(i))$$

$$\wedge$$

$$(\pi_i \wedge (motion_i = coherent) \wedge connected(i) \Rightarrow (turn90Move(i) \wedge \bigcirc motion_i = forward))$$

note: proposition π_i is true if robot i is taking an action

Overall swarm specification

Each robot must satisfy both Safety and Liveness properties at all future times

$$Robot_i := \Box(Safety_i \wedge Liveness_i)$$

Then specify the Swarm as the logical 'and' of all the robots

$$Swarm := Robot_1 \wedge Robot_2 \wedge ... \wedge Robot_N \wedge \Box(\pi_1 \oplus \pi_2 \oplus ...\pi_N)$$

Ensure that only 1 robot taking action at a time

Specification of Emergent Properties

Each robot is always connected

$$property1 := \Box\Diamond(\forall i \in robotSet.\ connected(i))$$

Eventually each robot will be connected to at least *k* distinct others

$$\begin{aligned}&property2 := \\ &\quad \Diamond\Box(\forall i \in robotSet. \\ &\qquad (\exists j_1 \in robotSet\{i\}.\ inRange(i, j_1) \wedge \\ &\qquad \exists j_2 \in robotSet\{i\}.\ inRange(i, j_2) \wedge \\ &\qquad \ldots \\ &\qquad \exists j_k \in robotSet\{i\}.\ inRange(i, j_k) \wedge \\ &\qquad distinct(j_1, j_2, \ldots, j_k)))\end{aligned}$$

First specify the emergent properties

Now attempt to prove (or disprove) that the swarm of robots satisfies the emergent behaviours

$$Swarm \Rightarrow property1$$

$$Swarm \Rightarrow property2$$

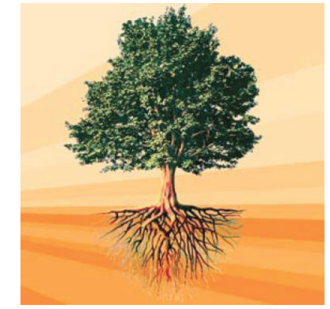
See: Winfield, Sa, Gago, Dixon and Fisher, IJARS 2 (4), 2005.

General conclusions

- It appears to be the case that we can develop approaches for verifying emergent swarm properties
  - mathematical modelling
  - formal logic verification and proof
- *but only for the ad-hoc approach*
- Does this mean that we must eschew the evolved approach?

Thank you!

EPSRC
Engineering and Physical Sciences Research Council



SEVENTH FRAMEWORK PROGRAMME

- Acknowledgements, colleagues in the BRL, but especially:
  - Dr Chris Harper, Dr Julien Nembrini, Dr Wenguo Liu, Dr Jan Dyre Bjerknes
  - Prof Chris Melhuish, Dr Tony Pipe

- Relevant publications:

  - Winfield AFT, Sa J, Gago MC, Dixon C and Fisher M, 'On Formal Specification of Emergent Behaviours in Swarm Robotic Systems', Int. J. Advanced Robotic Systems, 2 (4), 363-370, 2005.
  - Harper CJ and Winfield AFT, 'A Methodology for Provably Stable Behaviour-based Intelligent Control', Robotics and Autonomous Systems, 54 (1), 52-73, 2006.
  - Winfield AFT and Nembrini J, 'Safety in Numbers: Fault Tolerance in Robot Swarms', Int. J. Modelling Identification and Control, 1 (1), 30-37, 2006.
  - Winfield AFT, Liu W, Nembrini J and Martinoli A, 'Modelling a Wireless Connected Swarm of Mobile Robots', Swarm Intelligence, 2 (2-4), 241-266, 2008.
  - AFT Winfield, 'Foraging Robots', Encyclopedia of Complexity and Systems Science, Editor-in-chief: Robert A Meyers, Springer, 2010.