Herding Fish

From TUNA to CoSMoS

Adam Sampson
University of Kent

University of Kent | Computing

Introduction

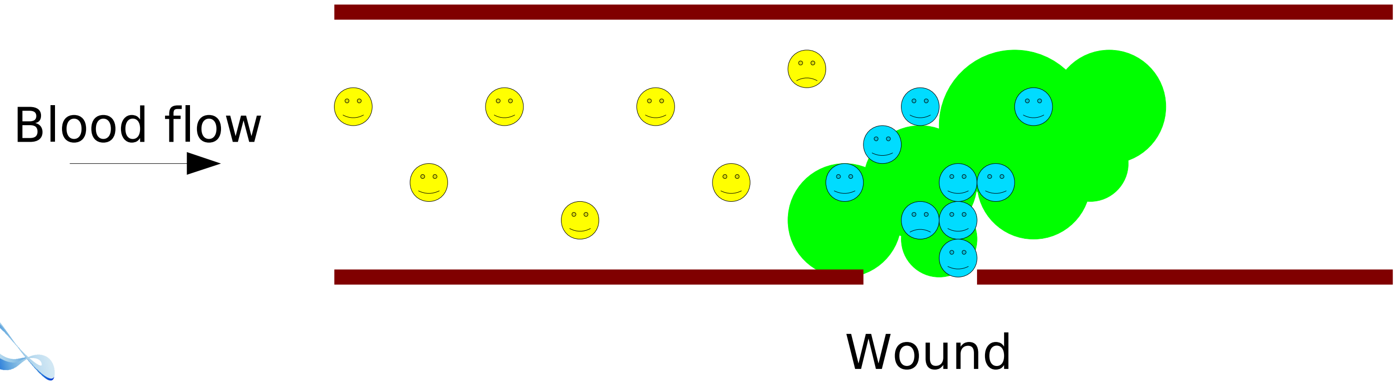
- Partly a wrap-up of TUNA
- Partly an advert for CoSMoS
- Will concentrate on simulation aspects
  - ... since that's most likely to be of interest to the people here

TUNA

- "Technology Underpinning Nanotech Assemblers"
- Two-year, £60k pilot project
- Studied the feasibility of engineering emergent behaviour in networks of nanites

Case study: haemostasis

- Artificial blood platelets
- Block wounds via an emergent behaviour:
  - Wounds emit chemical markers
  - Platelets become sticky in response
  - Platelets clump together and stick to edges of wound



University of Kent | Computing

Engineering emergent behaviour

- Emergent behaviour appears everywhere in nature – being able to design it would be really useful
- Break a problem down into processes that solve the problem as an emergent behaviour
  - i.e. refine emergent behaviour into simple behaviours
- But conventional refinement doesn't work...

Emergence does not refine

- No single refinement step between specification and implementation
  - Need intermediate environment
- See “Emergent properties do not refine” paper

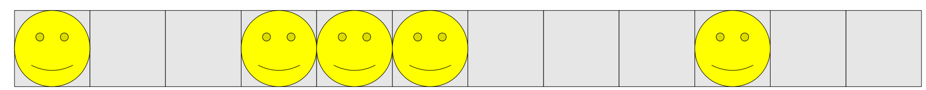
Modelling languages

- CSP-based approaches
  - Circus (CSP with Z)
  - CSP||B (CSP with B)
- Use FDR and Probe for static analysis
- Translate into real code for implementation
  - Initially by hand
  - Investigated compiling CSP directly

University of Kent | Computing

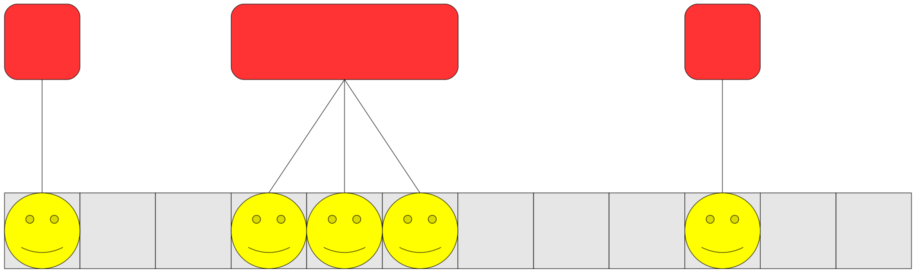
Blood clotting model

- Stochastic cellular automaton
  - Lots of states!
  - Clot movement requires coordination
- Modelled and verified using CSP
  - 1D, 2D and 3D models



Evolving the model

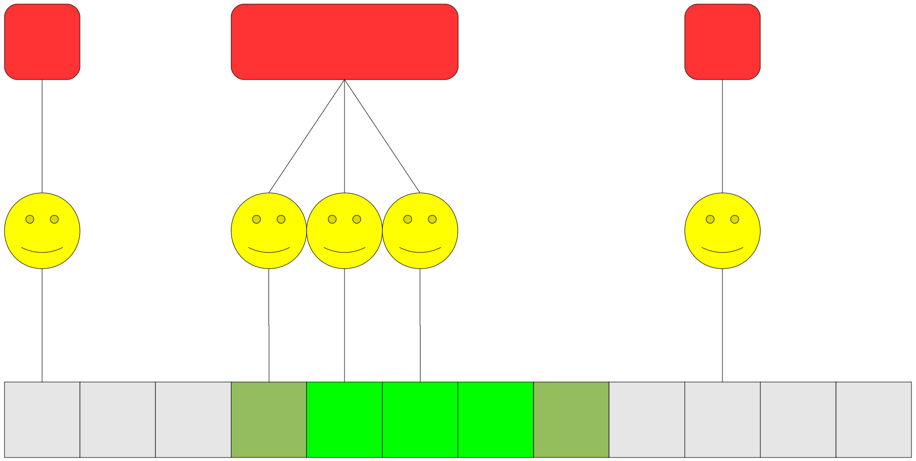
- Added extra layer of clots above platelets
  - Synchronises platelet movement
  - Clots merge when they collide
  - Behaviour is the same, but the rules have moved layers



University of Kent | Computing

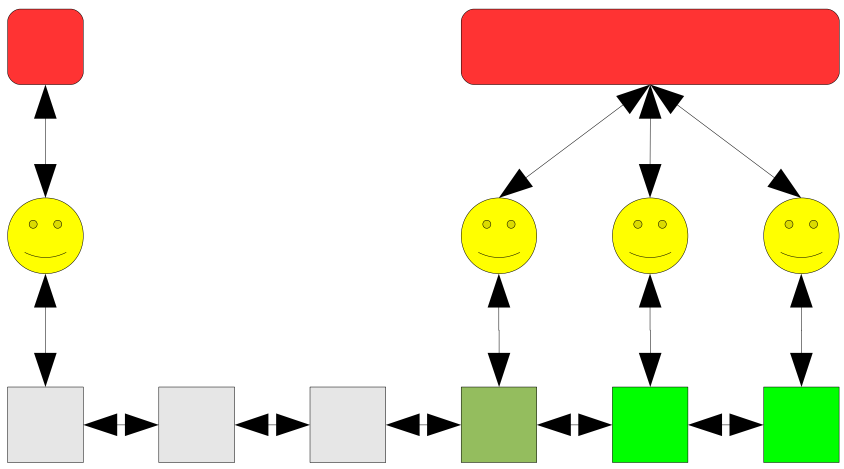
Evolving the model

- Environmental layer below platelets
  - Models chemical diffusion



Process-oriented programming

- Based on CSP and pi-calculus models of concurrency
- Isolated processes communicate via channels
- Everything is an agent!

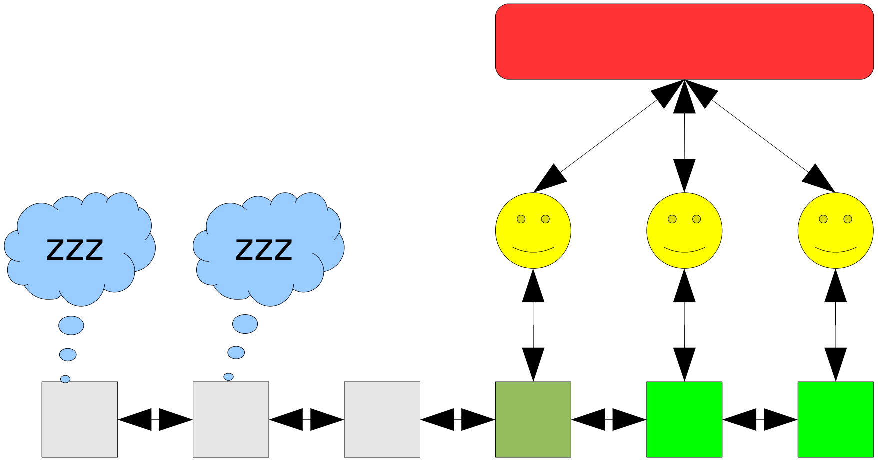


Languages and toolkits

- occam-pi programming language
  - Process-oriented
  - Mobile data provides efficient, safe data sharing
  - Supports millions of concurrent processes on commodity hardware
  - Takes advantage of multi-core processors
- Also JCSP, Handel-C, C++CSP, PyCSP…

Dependency tracking

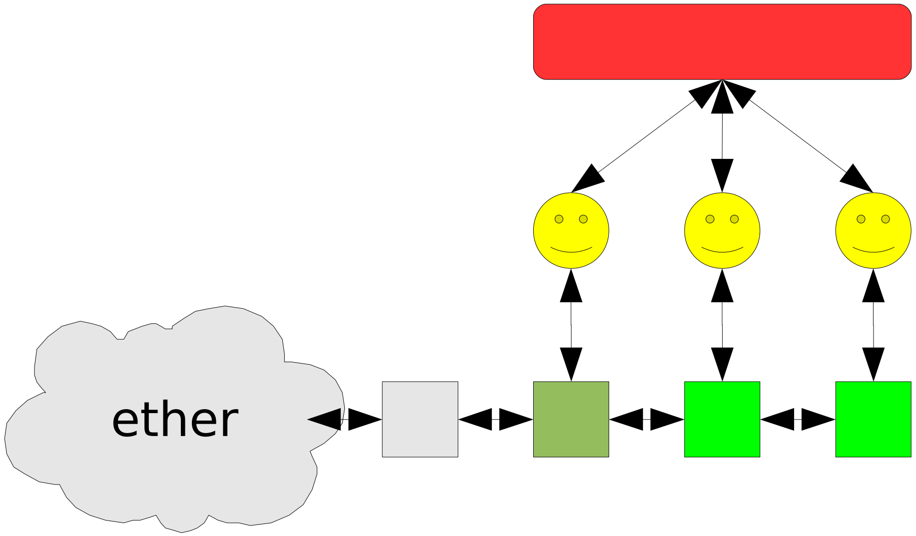
- Only recompute data when necessary
  - Process network provides dependency graph
  - Processes "sleep" when nothing is changing



University of Kent | Computing

Just-in-time world creation

- Maintaining empty world is expensive
- Create low level processes “just in time” in response to higher-level actions
- Allow unused processes to die



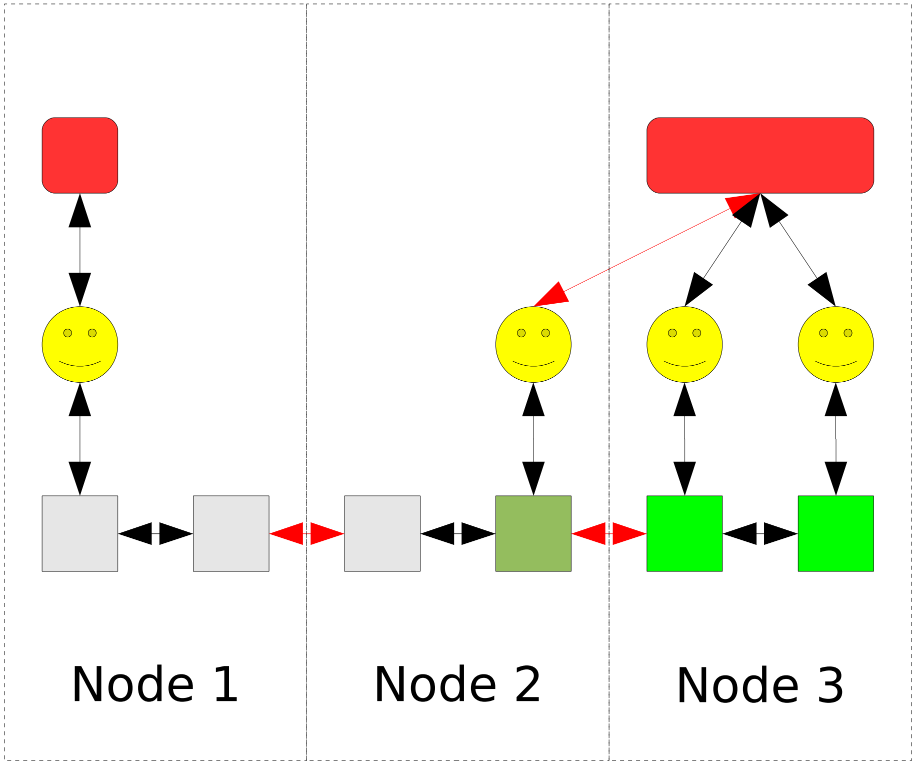
University of Kent | Computing

Distributing the model

- TUNA cluster
  - 40 Pentium 4 PCs with gigabit networking
  - Much cheaper than we expected!
- occam-pi provides transparent networked channels
  - Same semantics, but much higher latency

Distributing the model

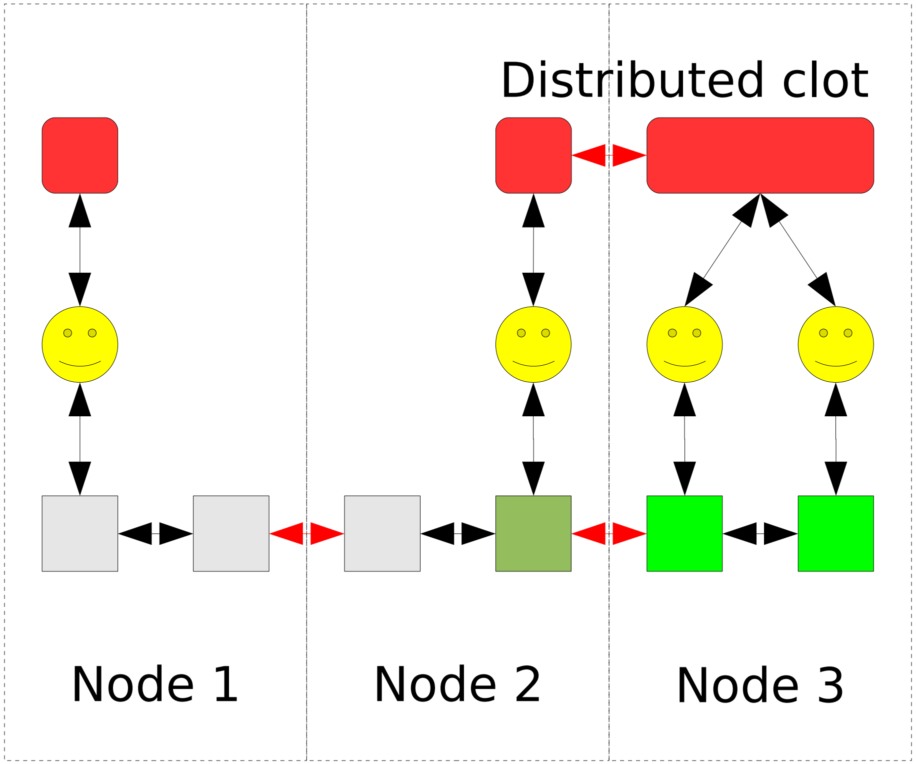
- Can simply split up the existing model
- Some links are now very high-latency, though...



University of Kent | Computing

Improving efficiency

- For better efficiency, make high-level processes aware of machine boundaries
- Bundle up expensive communications



University of Kent Computing

Blood clotting simulation

- Using the tricks above...
- Distributed across 24 nodes of the cluster
- Can do 40,000,000+ processes (at 1.5fps)
- (Nearly!) linear speedup as more nodes are added
- Visualisation based on VTK volumetric renderer

Blood clotting demo

CoSMoS

- “Complex Systems Modelling and Simulation”
- Four-year, £1.1m project
- Inspired by TUNA, but different focus
- Started October 2007

CoSMoS aims

- Design patterns and software framework for modelling and simulating complex systems
  - Applicable to many different types of systems
  - Provide a pattern language for modelling
  - Make efficient, massively-concurrent simulation accessible to scientists from other disciplines

People involved

- **York CS/EE/Chem**: Susan Stepney, Jon Timmis, Martin Bates, Fiona Polack, Andy Tyrell
- **Kent CS**: Peter Welch, Fred Barnes
- Christopher Alexander
- Microsoft
- Celoxica

Case studies

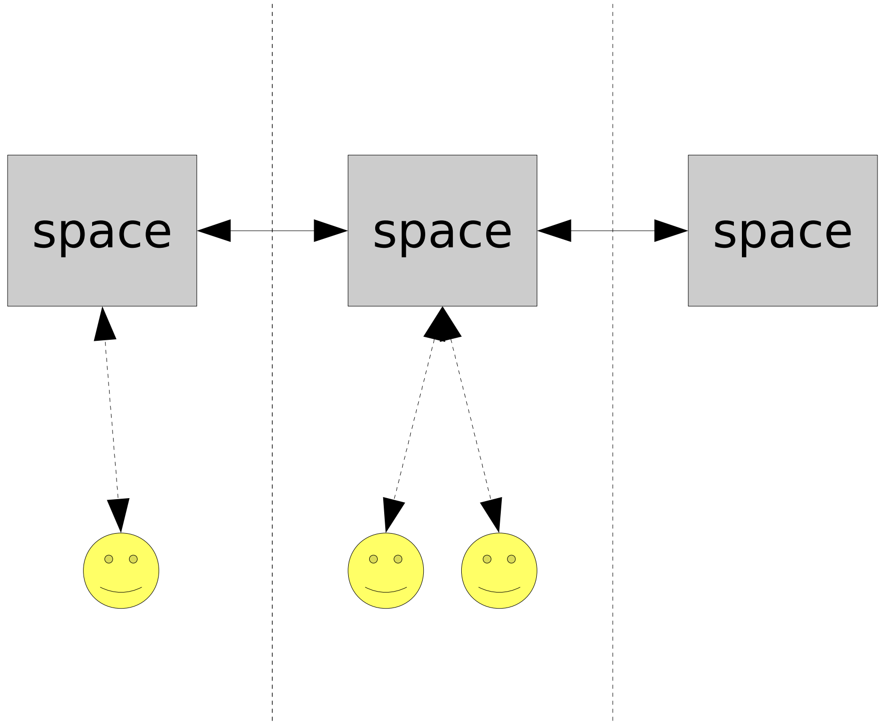
- Plants and animals – often grid-based
  - Fungi – ideal for multi-level modelling
- Liquid crystals
  - Continuous space, detailed physics model
- Bio-inspired algorithms, such as artificial immune systems
  - Common concepts between different bio-inspired systems

Distributed continuous space

- The first work we've done for CoSMoS...
- TUNA concentrated on discrete space
- Extend approach to continuous positioning
  - ... but take advantage of the TUNA techniques

Continuous model of space

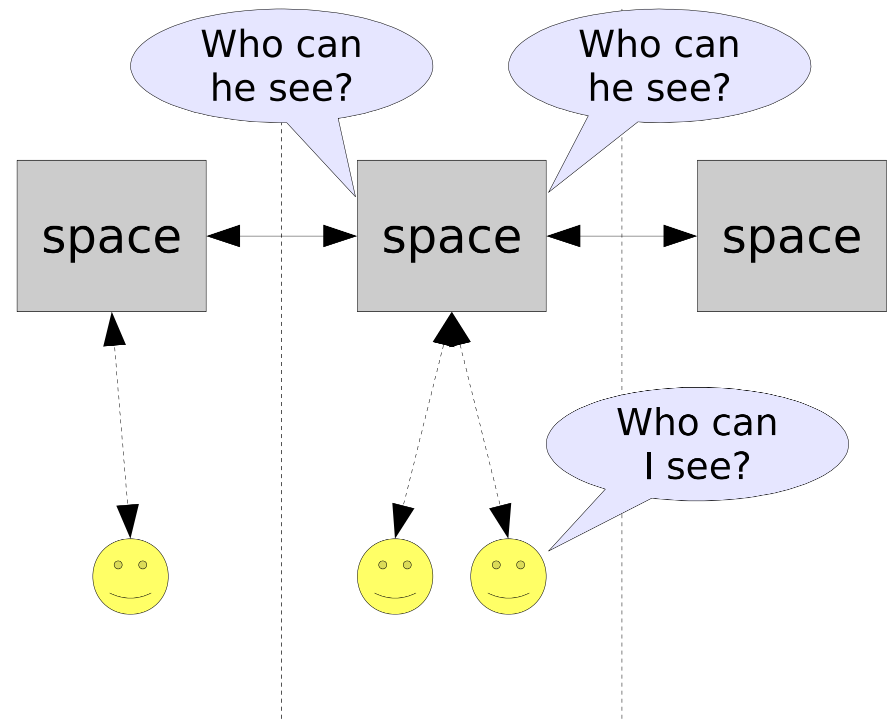
- Divide space into cell processes
- Space cells keep track of relative positions of local agents
- Hand off agents as they move between cells



University of Kent | Computing

Agent vision

- To see one cell in any direction, you need to ask the cells around you
- Vision requests are forwarded between cells



Boids

- Simple flocking rules
  - Average position
  - Average speed
  - Maintain minimum distance from others

Boids demo

Any questions?

This work supported by EPSRC grants
EP/C516966/1 and EP/E049419/1

http://www.cosmos-research.org/

First workshop will be at
Communicating Process Architectures 2008
September 2008 at York