

Computing Department

LANCASTER UNIVERSITY
University of Applied Sciences

Artificial Life – Ethology

CSc 355

Alan Dix
dixa@comp.lancs.ac.uk

Manolis Sifalakis
mjs@comp.lancs.ac.uk

Computing Department

Lecture Overview

- What is Alife
- Brief history – timeline
- Synthetic Ethology and Food Chains
- Example: Food Chain model
 - Agent Anatomy
 - (Pseudo-) code
 - Sample iteration
 - Results
 - Observations
- Reference List

Computing Department

What is Alife ?

- **Alife** [Langton] is set of mechanisms used to model and simulate evolving natural systems
 - Insect ecologies, animal behavior, negotiating entities, resource use in artificial economies
 - Studies the evolution of agents, or populations of computer simulated life forms in artificial environments
 - Complements traditional biology by trying to *recreate* biological phenomena

Computing Department

What is Alife ?

```

graph TD
    Alife --> CM[Computational Models  
(soft approach)]
    Alife --> Robotics[Robotics  
(hard approach)]
    Alife --> Biochemistry[Biochemistry  
(wet approach)]
    CM --> CA[Cellular Automata]
    CM --> NN[Neural Networks]
  
```


- Traditionally AI: a **top down** approach
- Alife: works from the **bottom up**

Computing Department

Brief history - Timeline

```

graph TD
    A["Jacques de Vaucanson  
(1739, pre-computer)  
Artificial Duck"] --> B["John Von Neumann  
(late 1940s)  
Theory of automata  
Self-replicating Machine"]
    B --> C["Homer Jacobson  
(1950s)  
Illustrated basic self-replication  
with a model train set"]
    C --> D["John Horton Conway  
(1960s)  
The Game of Life  
(Most famous cellular automaton)"]
    D --> E["Edgar F. Codd  
(1965)  
Simplified Von Neumann's Cellular  
automaton from 29 states to 8 states"]
    
    F["Christopher Langton  
(1979)  
First self-replicating computer organism  
using only an Apple II desktop computer  
(1989)  
Founder of Artificial Life"]
    G["Stephen Wolfram  
(1982)  
1-D Cellular Automata applied to natural  
phenomena (seashell patterns, plant growth)"]
    
    H["The Unit of Theoretical Behavioural Ecology  
Free University of Brussels  
(mid 1990s)  
Self-organization theories to  
research the behavior of social insects"]
    
    A --- F
    B --- F
    C --- F
    D --- F
    E --- F
    F --- G
    F --- H
  
```



Computing Department

Synthetic Ethology & Food Chains

- **Synthetic Ethology**
 - Study of animal behavior in which simple, synthetic organisms are allowed to behave and evolve in a synthetic world.
 - Branch of zoology
- **Food Chain**
 - Describes the hierarchy of living organisms within an ecosystem.

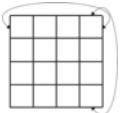
```

graph TD
    Carnivore -- consumes --> Herbivore
    Herbivore -- consumes --> Plant
  
```

Computing Department LANCASTER UNIVERSITY

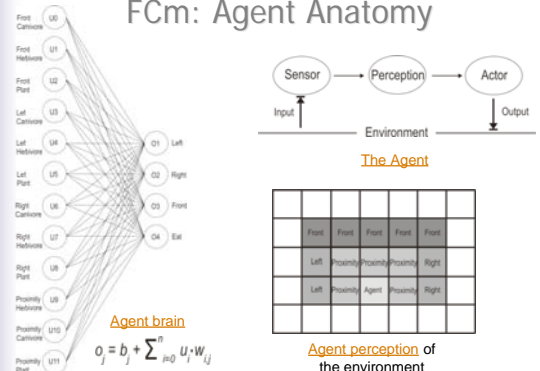
Example: Food Chain Model (FCm)

- 3 Entities
 - Plants:
 - Fixed location, consumed by herbivores
 - Herbivores:
 - Migratory agents, eat plants, eaten by carnivores
 - Carnivores:
 - Migratory agents, eat herbivores, die from starvation
- Environment
 - Toroid grid
 - Each cell occupied by one or more agents



Computing Department LANCASTER UNIVERSITY

FCm: Agent Anatomy



Sensor → Perception → Actor

Input ↑ Environment ↓ Output

The Agent

Agent brain

$$O_j = b_j + \sum_{i=0}^n U_i \cdot W_{ij}$$

Agent perception of the environment

Computing Department LANCASTER UNIVERSITY

FCm: Agent "Life & Death issues"

- Energy (E) / Metabolism
 - Eat $\rightarrow E = E + 1$
 - For each step $\rightarrow E = E - X$, ($H: X=1, C X=2$)
 - If $E == 0 \rightarrow$ Die
- Reproduction
 - If $E > 90\% \rightarrow$ Reproduce asexually
 - Lamarckian: Offspring inherits parents' NNet followed by random mutation of weights
- Death
 - Starvation (no food found)
 - Eaten (only for herbivores)

Computing Department LANCASTER UNIVERSITY

FCm: The (pseudo-) code

```

Main ()
  Init ()
  while (run < MAX_RUNS)
    SimulateOnce ()

Init ()
  landscape := InitLandscape ()
  GrowPlants ( landscape [plants] )
  while ( agents < MAX_AGENTS )
    agent := CreateAgent ()
    if ( agent.type == herbivore )
      PositionAgent ( landscape [herbivores] )
    else
      PositionAgent ( landscape [carnivores] )
  
```

Computing Department LANCASTER UNIVERSITY

FCm: The (pseudo-) code

```

SimulateOnce ()
  forall agent types
    foreach agent
      PerceiveEnvironment (agent)
      ForwardPropagateInputs (agent.Nnet)
      ComputeAction (agent)
    switch (agent.action)
      case TURN_LEFT:
      case TURN_RIGHT:
        agent.direction := UpdateOrientation (agent)
      case MOVE_FRONT:
        agent.position := UpdatePosition (agent)
      case EAT:
        Eat (agent)
  
```

Computing Department LANCASTER UNIVERSITY

FCm: The (pseudo-) code

```

...
  UpdateEnergy (agent, agent.action)
  if agent.energy == 0
    KillAgent (agent)
  else
    agent.age += 1
  if agent.energy > REPRODUCTION_LEVEL
    ReproduceAgent ( agent )
  
```

Computing LANCASTER UNIVERSITY

FCm: The (pseudo-) code

```

GrowPlants ()
  location := SelectRandomLocation (landscape [plants])
  if no plant in location
    landscape [plants] [location.x] [location.y] := 1

CreateAgent ()
  agent.energy := MAX_ENERGY / 2
  agent.age := 0
  agent.generation := 1
  agent.type := carnivore | herbivore
  foreach neuron in Nnet
    SetWeight (neuron)
  
```

Computing LANCASTER UNIVERSITY

FCm: The (pseudo-) code

```

PositionAgent ()
  location := SelectRandomLocation (landscape [agent.type])
  if no agent in location
    landscape [agent.type] [location.x] [location.y] := 1
  agent.direction := SelectRandomDirection ()

Eat ()
  if agent.type == CARNIVORE
    UpdateLandscape ( landscape [herbivores] )
  else
    UpdateLandscape ( landscape [plants] )
  
```

Computing LANCASTER UNIVERSITY

FCm: The (pseudo-) code

```

KillAgent ()
  UpdateLandscape ( landscape [agent.type] )
  if num of agent of this type < MAX_AGENTS / 4
    CreateAgent ()

ReproduceAgent ()
  if num of agents of type < MAX_AGENTS / 2
    // Inheritance of NNet in offspring
    new_agent := DuplicateAgent ( agent )
    // Randomly mutate neuron weights
    foreach neuron in new_agent.Nnet
      if mutation_probability > 50%
        SetWeight (neuron)
    PositionAgent ( landscape [agent.type] )
  
```

Computing LANCASTER UNIVERSITY

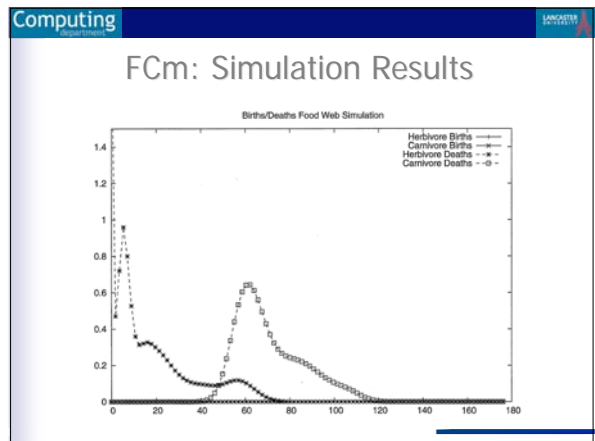
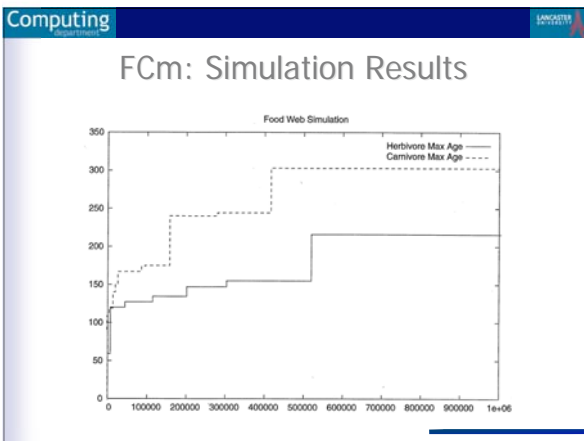
FCm: A sample iteration


Inputs = [0 1 0 0 0 0 0 0 0 0 0 1]
 outputs left=1 right=-1 front=0 eat=1

Inputs = [0 1 0 0 0 0 0 0 0 0 0 0]
 outputs left=1 right=0 front=0 eat=0

Inputs = [0 0 0 0 0 0 0 0 1 0 0 0]
 outputs left=1 right=0 front=1 eat=0


b (left)	= [1]	w (left)	= [1 0 0 0 0 1 0 0 0 0 0 0]
b (right)	= [0]	w (right)	= [0 0 0 0 -1 -1 0 0 0 0 0 -1]
b (front)	= [1]	w (front)	= [0 -1 0 0 1 0 0 0 0 0 0 0]
b (eat)	= [0]	w (eat)	= [1 0 0 0 0 -1 -1 0 0 0 0 1]



Computing 


FCm: Observations and Conclusions

- Competition
 - Carnivores evolve NNets, good at **locating and eating herbivores**
 - Herbivores evolve NNets that **find plants and avoid carnivores**
- Evolved Strategies
 - **Herding**: Herbivores follow other herbivores in front
 - **Ambushing**: Carnivores find plants and then wait for herbivores to wander by

Computing 


FCm: Observations and Conclusions

- Parameters tuning
 - Number of **plants** \geq number of **herbivores**
 - Number of agents must be **small** so as not to crowd the simulation
 - Number of **carnivores** $\leq 2 * \text{Number of herbivores}$

Computing 

Reference List

- Seminal paper
 - Christopher G. Langton. **Artificial Life**. Proceedings of interdisciplinary workshop on the Synthesis and Simulation of Living Systems, Los Alamos, 1987. Addison-Wesley. 1989
- Zooland: "The Artificial Life Resource"
 - <http://surf.de.uu.net/zooland/>
- Book chapter on Artificial Life
 - M. Tim Jones. 2003: **AI Application Programming**. Charles River Media, Inc.

Computing 

Questions ...