Self-organisation and Emergence in Multi-Agent Systems

Giovanna Di Marzo Serugendo
dimarzo@dcs.bbk.ac.uk

Birkbeck College, University of London
http://www.dcs.bbk.ac.uk/~dimarzo
Outline

• Context
  – ESOA Workshops
  – AgentLink Self-Org TFG

• Definitions
  – Self-organisation / Emergence

• Overview
  – Engineering:
    • Agents / Self-organising Mechanisms / Middleware Infrastructure / Methods and tools

• Applications
  – Manufacturing control (ants)
  – Region detection (spiders)
  – P2P protocols (gossip)
  – Co-field middleware infrastructure (force fields)
  – Adelfe method (cooperation)
Context

• Engineering Self-Organising Applications (ESOA workshops)
  – 2003 to 2006
  – Co-located with AAMAS
  – Papers
    • Applications
      – MANETS, manufacturing control, traffic control, …
    • Self-organising mechanisms
      – Ants-based, immune system, …
    • Methodologies/Software Engineering
      – Middleware, methods, design patterns, …
Context

• Agentlink Technical Forum Group on Self-Organisation in MAS (Self-Org TFG)
  – 4 meetings between 2004 and 2006
  – Continues once per year
    • Since 2006, Co-located with EUMAS workshop
  – Goal:
    • Enhance Interdisciplinary links
    • Work on concepts / definitions
    • Produce a book
  – Achievements
    • Some operational definitions (!) / characteristics
    • Series of papers and reports
    • Book proposal
  – http://www.irit.fr/TFGSO
Self-Organisation

• Self-organisation in Engineered systems

“Self-organisation is the process enabling a system to change its organisation in case of environmental changes without explicit external command.”

“Strong self-organising systems are those systems where there is re-organisation with no explicit central control, either internal or external.”

“Weak self-organising systems are those systems where, from an internal point of view, there is re-organisation under an internal central control or planning.”

– Dimarzo et al. (2005)
Emergent Phenomenon

- Emergent phenomenon in engineered systems

“Emergent phenomenon is a functionality, structure/organisation, characteristics or property of a system not explicitly coded in the local components, visible by an observer at the macro-level but not necessarily at the micro-level.”

“Weak emergent phenomena are those phenomena we can define as being the result of a (complex) operational function $F(\text{components})$.”

“Strong emergent phenomena are those phenomena we cannot define as the result of an operational function or $F$ is ‘analytic’.”

- Lisbon Self-Org TFG Meeting – Draft - 2006
Overall Picture

Natural Self-organising Systems

Self-organisation Mechanisms

Artificial Self-organising Systems

Analysis and Simulation

Engineering Self-organising Systems
Engineering Overview

Self-organising Mechanisms

Agents

Middleware Infrastructure

Open Issues

Methodologies Tools
Engineering

• Software Agents
  – Low level active components

• Self-Organising Mechanisms
  – Interactions / coordination among agents

• Middleware Infrastructures
  – Support for
    • Agent execution and interactions
    • Self-organising Mechanisms

• Methods/Methodologies and tools
Agents

• Software components
  – **Autonomous** actions
  – Situated in an environment
    • Sensors / Actuators
  – Social behaviour
  – Interactions / coordination
  – Intelligent / mobile

• Interest of agents for self-organisation
  – Naturally play the role of the individual **“building blocks”** in a self-organising system
    • Agents can be ..
      – … ants
      – … services
      – … peers / nodes
      – … traffic lights
      – … cars
# Self-Organising Mechanisms

<table>
<thead>
<tr>
<th>Metaphor</th>
<th>Description</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stigmergy</td>
<td>Ants</td>
<td>Manufacturing Control / MANETs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Decentralised Car Traffic Control</td>
</tr>
<tr>
<td></td>
<td>Spiders</td>
<td>Region Detection</td>
</tr>
<tr>
<td></td>
<td>Bees</td>
<td>Cultural Heritage</td>
</tr>
<tr>
<td>Social Human Behaviour</td>
<td>Trust and Reputation</td>
<td>Security / Access Control</td>
</tr>
<tr>
<td></td>
<td>Gossip</td>
<td>P2P Protocols / Overlay Networks</td>
</tr>
<tr>
<td></td>
<td>Tag-based</td>
<td>Specialisation and P2P Protocols</td>
</tr>
<tr>
<td>Biological Systems (cells)</td>
<td>Immune System</td>
<td>Detection Intrusion and Response</td>
</tr>
<tr>
<td>Learning</td>
<td>Reinforcement Learning</td>
<td>Decentralised Traffic Light Control</td>
</tr>
<tr>
<td>Cooperation</td>
<td>AMAS</td>
<td>Flood forecasting / Robots simulations</td>
</tr>
<tr>
<td></td>
<td>DIET Platform</td>
<td>Lookup system for P2P networks</td>
</tr>
<tr>
<td>Architecture based</td>
<td>Holons</td>
<td>Enterprise Management</td>
</tr>
<tr>
<td></td>
<td>Metadata</td>
<td>Self-reconfiguration</td>
</tr>
<tr>
<td></td>
<td>Self-organising</td>
<td>Autonomic Computing</td>
</tr>
<tr>
<td></td>
<td>Architectures</td>
<td>Ambient Intelligence</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Middleware Infrastructures

• Coordination Spaces
  – “Blackboard”
    • Repository of tuples (n-uples) accessed concurrently
    • Producers and consumers of tuples
    • Associative memory
      – Retrieval of tuples based on “pattern matching”

• Swarm-Based Infrastructures
  – SwarmLinda [Tolksdorf 03]
  – AntHill [Babaoglu 02]

• Field-Based Infrastructures
  – Co-Fields [Mamei 02]
  – TOTA [Mamei 03]
Methodologies

• Adelfe (AMAS) [Gleizes]
  – Methodology based on the AMAS *cooperative* theory among agents

• Design Patterns [DeWolf]
  – Decentralised coordination patterns as design patterns
    • Gradient field / Market-Based Control

• Self-Organising Architectures
  – Meta-models of architectures (configuration)
  – Constraints of reconfiguration and interactions
Applications

- Manufacturing control (ants)
- Region detection (spiders)
- P2P protocols (gossip)
- Co-field middleware infrastructure (force fields)
- Adelfe method (cooperation)
Manufacturing Control

- Metaphor: Ant foraging
- PROSA Architecture [Hadeli 03]
- Agents:
  - Orders agents (logistics for managing products), products agents (processes tasks), resources agents (raw material, machines, etc)
- Mapping of control and production system into agents
  - Actual production system is reflected into an agents structure
  - Each resource/product/order has a corresponding resource/product/order agent (local information only)
  - Links among agents (e.g. order agents know about location of resources agents and products agents necessary to complete order)
  - Agents creates ant-agents (mobile agents) that explore the cyber production system and deposit/sense pheromone
Manufacturing Control

• Ant-agents behaviour
  – Feasibility information ants
    • Information related to the resource locations (availability, etc.)
  – Exploring ants
    • Order agents create several ants each exploring a way of realising the order (processing times, etc. gives back a report with followed route)
  – Intention propagation ants
    • Order agents create ants that propagate information about the order’s intentions (chosen best route). Ant has a fixed route, and makes bookings.

• Manufacturing control
  – Obtained from the choices made by order agents
    • On the basis of the above information
  – Actually executed by resources agents
Manufacturing Control

• Exploring Ants (EA)
  – Tries to find solutions
  – Searching for solutions is guided by local pheromones
  – Reports result of solution to the corresponding Order Agent
Region Detection

• Metaphor: Social Spiders
  – Few species of spiders are “social”
    • Sharing of web
    • Collaboration (preys, web weaving)
    • Stigmergy based on silk
      – Spiders follow silk or move to points where silk is fixed
Region Detection
Region detection

• Region detection (grey levels) [Bourjot 03]
  – Partition of image into subsets of separate objects
  – Determination of sets of connected pixels (regions)

• Idea:
  – Webs weaving determines the region

• Algorithm
  – Spider has to detect a given region (grey level)
    • Several spiders explore image and fix silk on relevant pixels
    • Silk attraction
    • Resulting web is fixed on “interesting” pixels
Region Detection
Region Detection
P2P Protocols

• **Metaphor: Gossip**
  – Light informal conversation for social occasions [WorldReference.com dictionary]
  – Periodic exchange and update of information among members of a group
  – Allows: aggregation of global information inside a population, social learning
  – Parameters: neighbourhood, level of precision of information

• **T-Man Algorithm [Jelasity 05].**
  – Generic protocol based on gossip communication model
  – Goal: network topology management problem
    • Nodes randomly connected
    • Re-organisation of connections to produce desirable topology
      – Nodes *choose* their behaviour
      – Nodes become neighbours based on information such as: geographic position, content, storage capacity
P2P Protocols

• Principle
  – Nodes maintain local view (profile) of neighbours
  – Ranking function defines the target topology (e.g. distance)
    • Serves for reorganising the set of neighbours
    • Based on “profile” of the nodes (e.g. number, useful info for topology)
  – Gossip message exchange
    • Choice of « closest » neighbour based on ranking function
    • Local exchange / combination of neighbours profile
    • Nodes become closer and closer
  – Allows adaptation of neighbours list
  – Re-organisation of the network topology

• Applications
  – Overlay networks supporting P2P systems
    • Maintenance or establishment of P2P topology
  – Sorting, Clustering of nodes, Distributed Hash table
P2P Protocols

after 3 cycles

after 5 cycles

after 8 cycles

after 15 cycles
Field-Based Infrastructure

• Co-Fields (Computational Fields) [Mamei 02]

• Principle – Force Fields
  – Agents generate application-specific fields
  – Propagation of fields in environment according to field-specific laws
  – Composition of different fields (coordination field)
  – Agents follow field gradient (downhill / uphill)
  – Agents movements are driven by fields (no central control)
  – Coordination emerges from
    • Interrelated effects of agents following the fields
    • Dynamic fields reshaping due to agents movements
    • Composition of different fields at each point
Field-Based Infrastructure

• Co-Fields Modelling of Ants Foraging [Mamei 02]
  – Two fields: Home and Food fields
    • Generated and spread by environment
  – Ants follow home or food field
  – Environment change fields according to ants movements
    • Wrinkling of fields where ants are located
    • Wrinkle = Abstraction for the pheromone
  – Fields = channels
    • Food-fields: down to food
    • Home-fields: down to home
  – Pheromone evaporation
    • Environment removes the wrinkle after elapsed time
Field-Based Infrastructure

- **TOTA – Tuples on the Air [Mamei 03]**
  - Based on Coordination Space
  - Uncoupled adaptive interactions
  - Provides Context-awareness
  - Follows Co-Field principle

- **TOTA System**
  - Agents *inject tuples* in the system
  - Environment *propagates* and *diffuses tuples* in the system
    • Propagation follows a specified pattern or propagation rule
  - Agents *locally sense* the resulting fields

- **Application Development**
  - Inject tuples (content + propagation rule)
  - Query *local* tuples (pattern-matching)
Methodologies

• ADELFE Methodology
  – Guide + help designer
  – Determination of type of system
  – Based on AMAS Theory
    • Cooperative agents
Cooperative Agent

• Cooperative attitude of an agent
  – Local and autonomous
  – Independent of the global function of the system
  – Heuristic to move through state space in a right direction

• Definition of cooperation (see types of NCS)
  – All perceived signals must be understood without ambiguity
  – Received information is useful for the agent’s reasoning
  – Reasoning leads to useful actions towards others agents
Cooperative Agent

• Cooperative Agents
  – **Skills** (what agent is able to do)
  – **Knowledge** about world (itself, other agents, environment)
  – **Interaction language**
  – **Aptitude** (reasoning)
  – **Social attitude** (cooperation)

• Cooperative agent fundamental activities:
  – **Perceives, decides and acts** in the world
    • If in a **cooperative** situation ➔ **realises** its function
    • If in an **uncooperative** situation (failure) ➔ **acts to come back** in a cooperative state
ADELFE Methodology

• Requirements
  – Definition of the studied system
  – Environment model:
    • agents, context, environment

• Analysis
  – Identification of agents
  – Adequacy to AMAS theory

• Design
  – Agent model
    • Cooperative agent design
  – Non cooperative situation model
    • Cooperative and non cooperative interactions
ADELFE Methodology

• Non cooperative situation (NCS) model
  – Table of all NCS cases
  – For each agent:
    • Agent state, NCS description, conditions, actions

• Three kinds of NCS
  – Signal perceived from environment is not understood (or with ambiguity)
  – Perceived information does not lead to activity
  – Actions are not useful for environment
Emergent Programming

• Self-assembly of instructions [Georgé 2005]

• Instruction-Agents
  – Interactions among agents
    • Send / receive data (input/output)

• A Program
  – One organisation of the instruction-agents

• Final program
  – Obtained by successive re-organisations (adaptations)

• Example: 6 agents
  – + agent, * agents
  – 3 constant-agents (A=2, B=10, C=100)
  – Output-agent (provides feedback from outside)
Emergent Programming

• Non Cooperative Situations
  – Agent is missing a partner for one of his inputs
    • Solution
      – Contact agent with output values of corresponding type
  – Agent is informed that another agent is in a missing situation
    • Solution
      – Try to act as the missing agent
      – Contact another agent

• Feedback from environment
  • Bigger (if highest value has to be produced) or
  • Smaller (if lowest value has to be produced)
Emergent Programming

\[ A \times B \times C \]

\[ 2 \times 10 \times 100 \]

\[ 200 + 210 \]

\[ 410 \]

OUTPUT
Engineering: Open Issues

• Considered systems
  – Open dynamic systems
  – Heterogeneous agents
  – Selfish or cooperative

• Issues
  – Interactions among:
    • Independently developed heterogeneous agents
  – Management of uncertainty
  – Design and development
    • Micro- / Macro-behaviour
    • Prediction of good/bad behaviour
  – Control
Additional Activities

• IEEE SASO Conference

• Agentlink Technical Forum on Self-Organisation
  – http://www.agentlink.org/activities/al3-tf

• ACM Transactions on Autonomous Adaptive Systems
  – Inaugural issue: September 2006
  – http://www.acm.org/pubs/taas

• IEEE ETTC Organic Computing Task Force
Papers

- Journals


Papers

• Agentlink Newsletters


Papers

• Books


Adaptation Mechanisms

• Translation of Natural Mechanisms
  – Stigmergy
    • Indirect communication through the environment
    • Digital pheromone
      – intensity, evaporation rate
    • Work-in-progress
      – mapping table: configurations – action

  – Gossiping
    • Informal discussion among entities
    • Local exchange of information (neighbours list)
Adaptation Mechanism

• Translation of Natural Mechanisms
  – Trust
    • Human trust in peers
    • Trust values, calculation of risk, decision of actions
    • Updated on basis of positive/negative evidence
  – Immune System
    • B cells + antibodies (detection and marking)
      T cells (destruction)
    • Bit strings (anomaly to detect)
    • Mobile agents (B and T cells)
Adaptation Mechanism

- Implementation of Artificial Mechanisms
  - Tags
    - Markings attached to individuals (agents) and observable by others
    - Agents change behaviour on basis of utility function value observed in peers (tag)
  - Metadata
    - Additional information (metadata) about functional / non-functional information / policies
    - Middleware processes metadata and components adapt to policies
Field-Based Infrastructure

- Principle – Force Field Metaphor
  - Propagation of tuples is similar to propagation of fields in the physical space
  - Particle do not interact directly but locally perceive the fields

- Implementation
  - P2P Network of (Mobile) nodes (running TOTA middleware)
  - Nodes maintain a limited list of neighbours
  - TOTA-tuple = content + propagation rule + maintenance rule
    - Content = information
    - Propagation rule
      - How to diffuse the tuple
      - Scope (distance) of propagation of the tuple
      - Direction of propagation
      - How to change tuple content during propagation
    - Maintenance rule
      - How tuple reacts to changes in its environment
  - TOTA Middleware actively supports tuples propagation
    - If new node join the system, tuples are propagated to this new node (according to their propagation rules)
Field-Based Infrastructure

Field Tuple

C = (id, distance)
P = (propagate everywhere, increment distance by one at each hop)
M = (update structure upon network topology changes)

Agents follow gradients
• Flocks of Birds
• Traffic Management
Field-Based Infrastructure

Pheromone Tuple

\[ C = (\text{id, strength}) \]
\[ P = \text{(propagate in neighbourhood)} \]
\[ M = \text{(evaporate by diminishing strength periodically)} \]

Agents sense pheromone

- Routing
Robots Simulation

- **Robot**
  - Autonomous
  - Resource transportation task
  - Internal State guided
  - Micro-level entity

- **Environment**
  - Two rooms
  - Narrow corridors separate the rooms
  - \( \rightarrow \) spatial interference

- Emergence of a traffic direction [Picard, 2002, 2005]