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

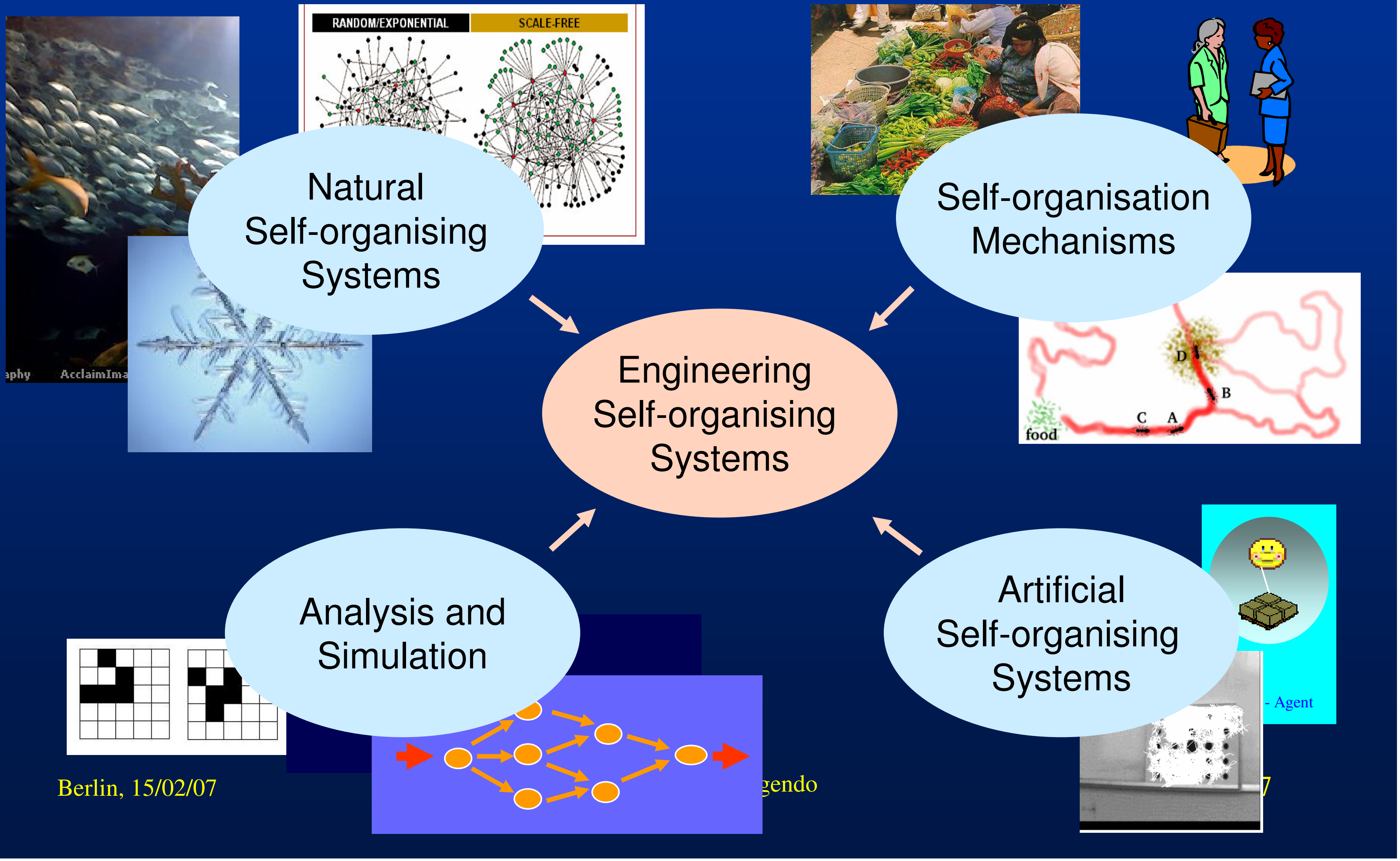
  “Emergent phenomenon is a functionality, structure/organisation, characteristics or property of a system not explicitly coded in the local components, visible by an observer at the macro-level but not necessarily at the micro-level.”

  “Weak emergent phenomena are those phenomena we can define as being the result of a (complex) operational function F(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

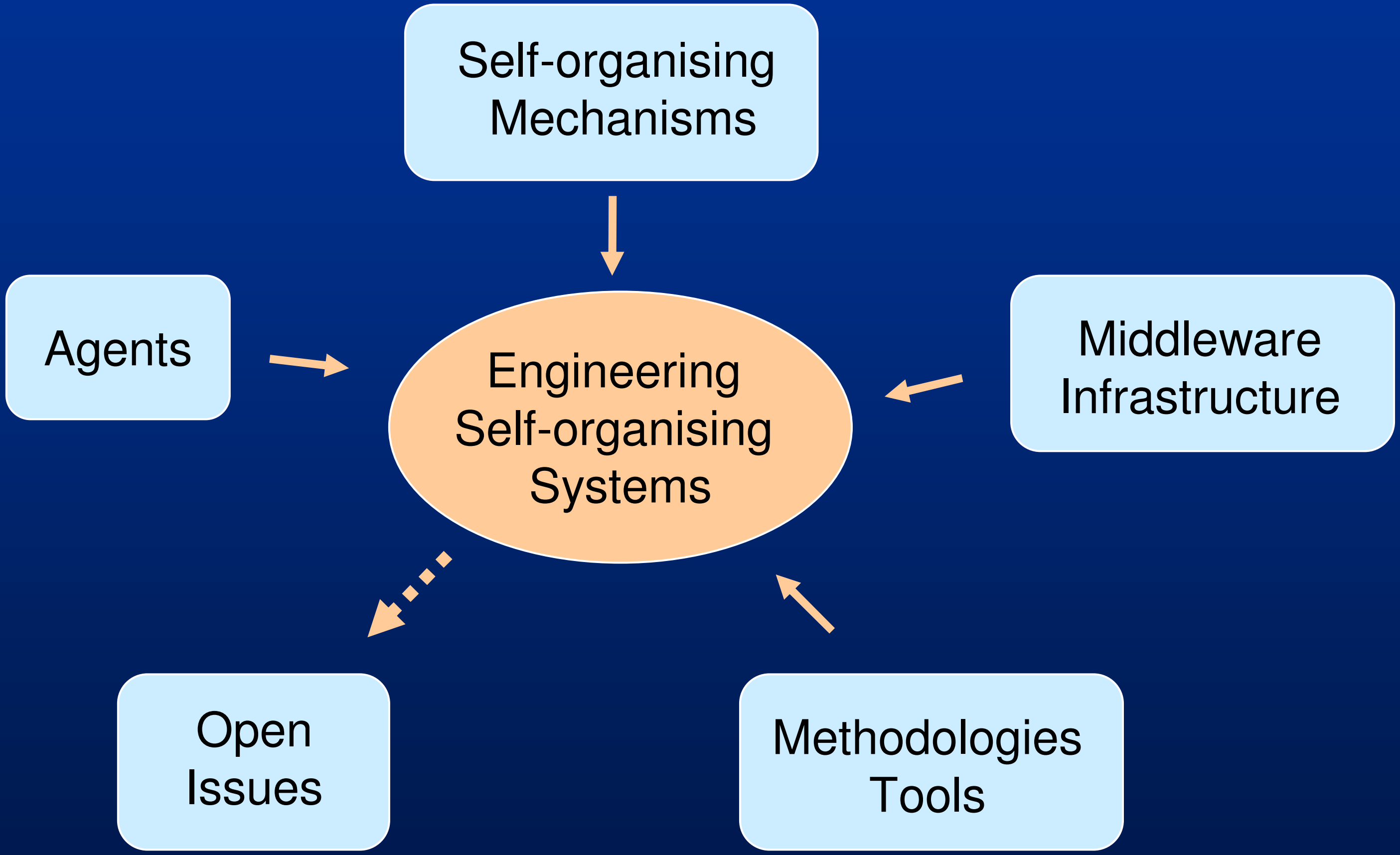
Self-organisation Mechanisms

Engineering Self-organising Systems

Analysis and Simulation

Artificial Self-organising Systems

Engineering Overview



Self-organising Mechanisms

Agents

Engineering Self-organising Systems

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

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

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

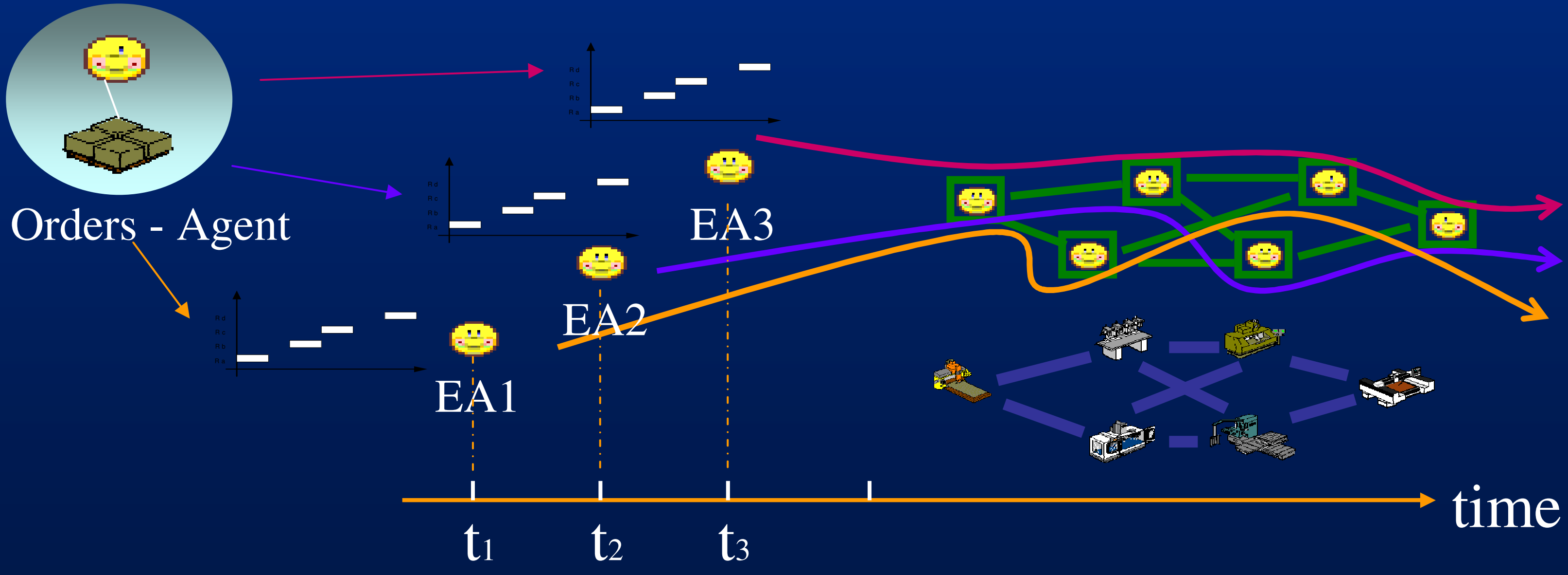
Manufacturing Control

- 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

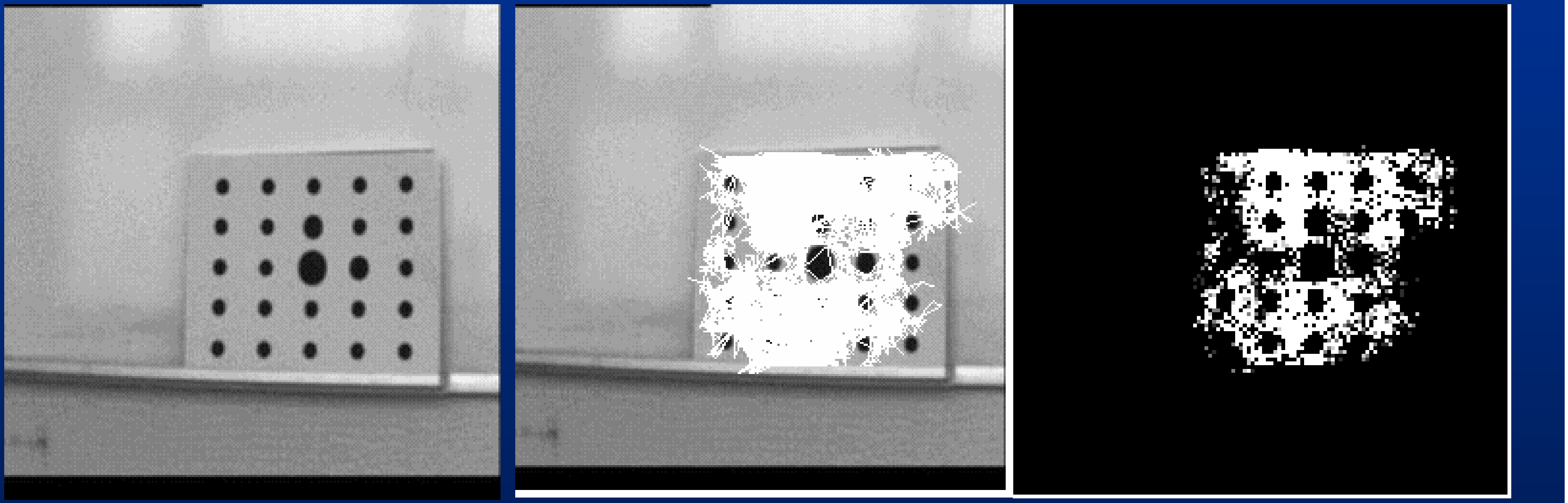
Region detection

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

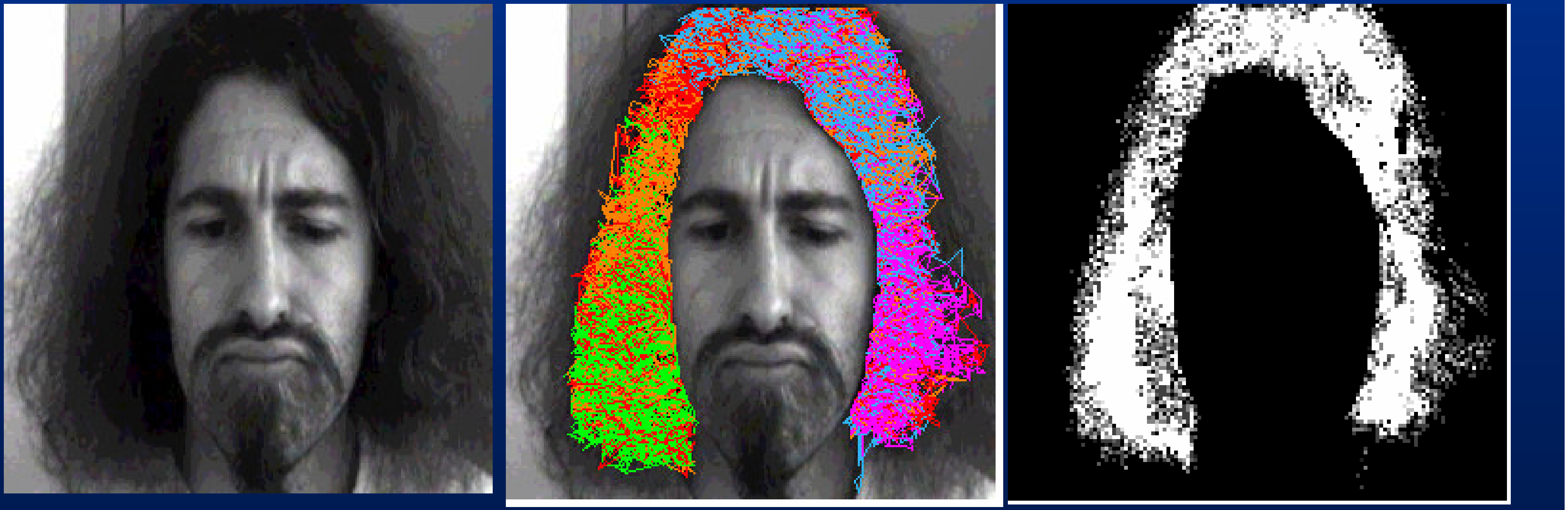
- Idea:
  - Webs weaving determines the region

- 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

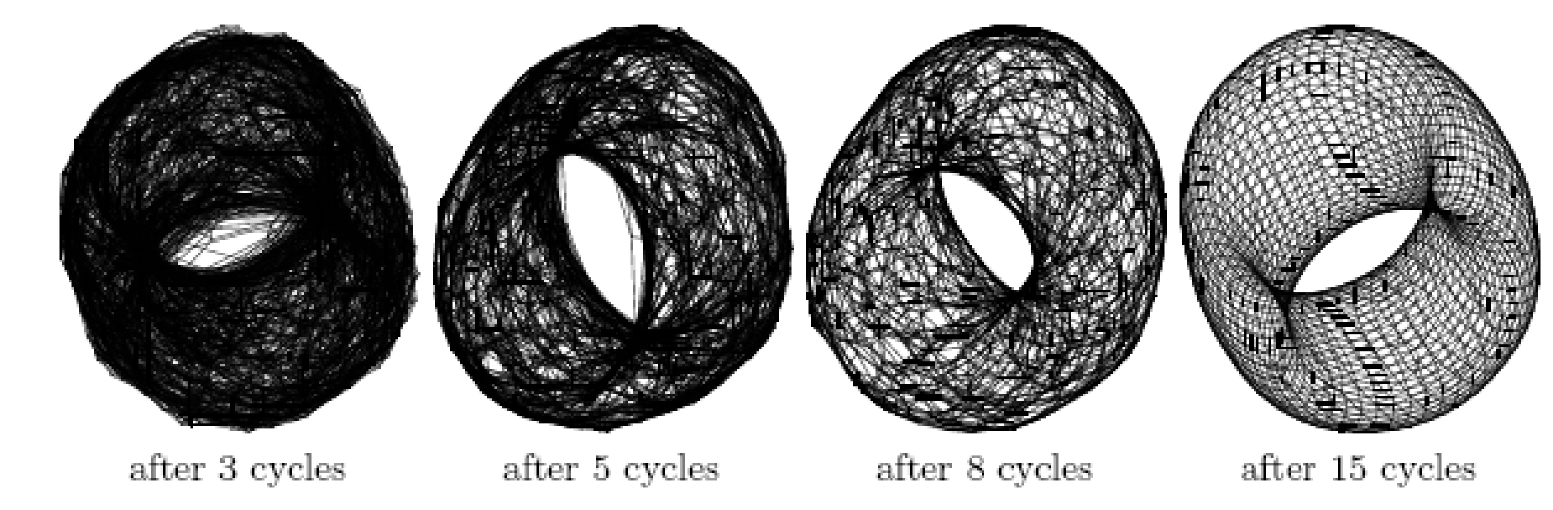
- T-Man Algorithm [Jelasity 05].
  - Generic protocol based on gossip communication model
  - Goal: network topology management problem
    - Nodes randomly connected
    - Re-organisation of connections to produce desirable topology
      - Nodes choose their behaviour
      - Nodes become neighbours 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

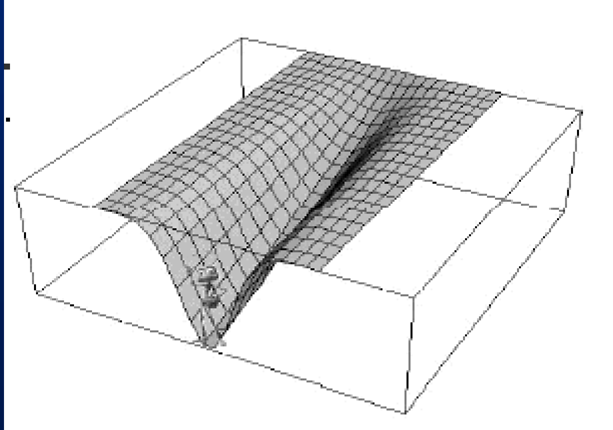
Field-Based Infrastructure

- 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

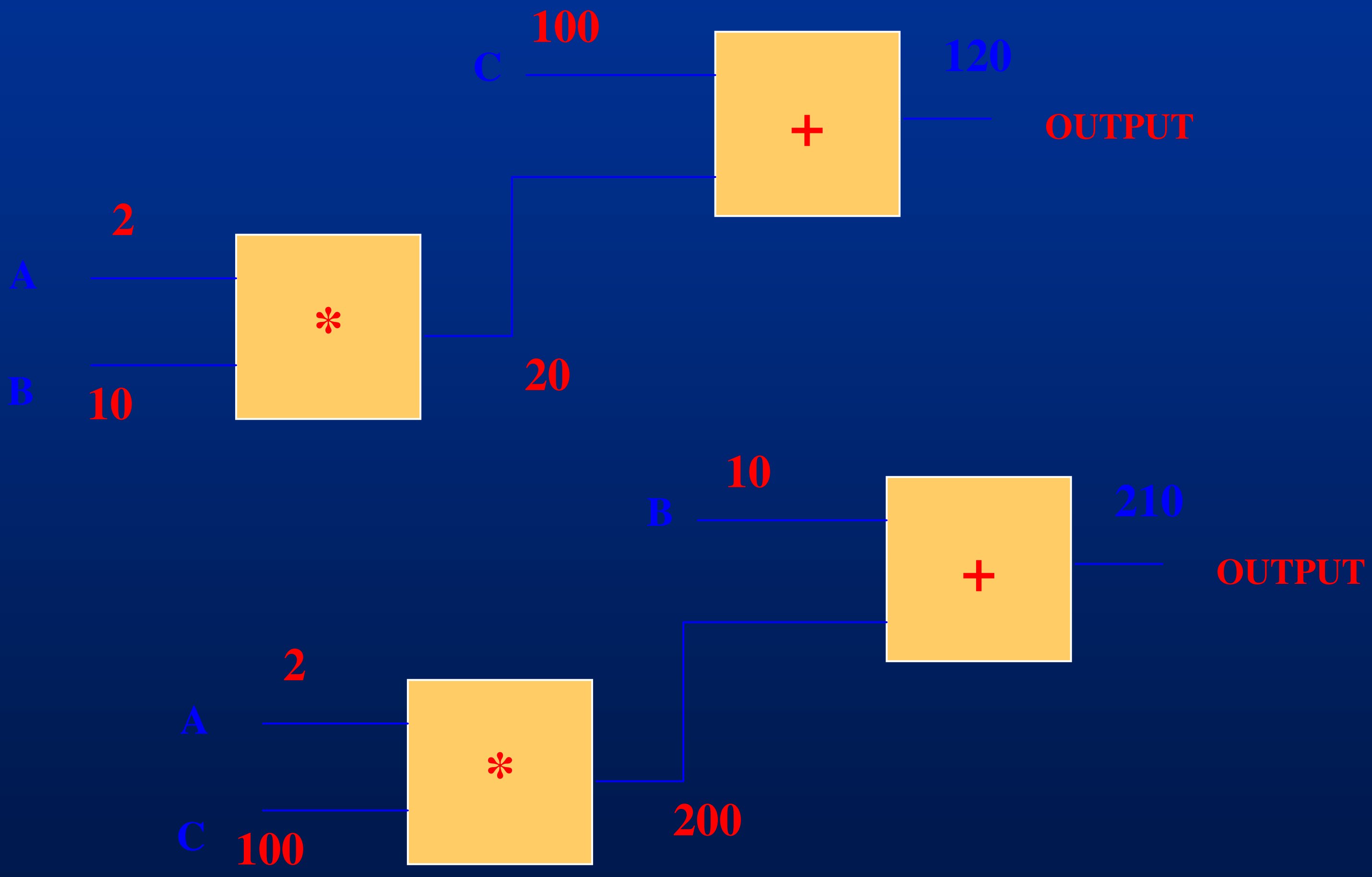
- Final program
  - Obtained by successive re-organisations (adaptations)

- 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



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
  - http://projects.csail.mit.edu/saso2007
- 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
  - http://www.neuroinformatik.ruhr-uni-bochum.de/PEOPLE/igel/oc.html

Papers

- Journals

- G. Di Marzo Serugendo, M.-P. Gleizes, A. Karageorgos. "Self-organisation and emergence in MAS: an overview", Informatica 30(1): 45-54, Slovene Society Informatika, Ljubljana, Slovenia, 2006.

- J.-P. Mano, C. Bourjot, G. Lopardo, P. Glize: Bio-inspired Mechanisms for Artificial Self-organised Systems, Informatica 30(1): 55-62 , Slovene Society Informatika, Ljubljana, Slovenia, 2006.

- S. Hassas, G. Di Marzo Serugendo, A. Karageorgos, C. Castelfranchi. "Self-organising mechanisms from social and business/economics approaches", Informatica 30(1):63-71, Slovene Society Informatika, Ljubljana, Slovenia, 2006.

- C. Bernon, V. Chevrier, V. Hilaire, P. Marrow: Applications of Self-Organising Multi-Agent Systems: An Initial Framework for Comparison, Informatica 30(1): 73-82 , Slovene Society Informatika, Ljubljana, Slovenia, 2006.

- G. Di Marzo Serugendo, M.-P. Gleizes, A. Karageorgos. "Self-Organisation in MAS", Knowledge Engineering Review 20(2):165-189, Cambridge University Press, 2005.

Papers

- Agentlink Newsletters
- G. Di Marzo Serugendo, Marie-Pierre Gleizes, Anthony Karageorgos "AgentLink Third Technical Forum Group Self-Organisation in Multi-Agent", AgentLink Newsletter, Issue 19, ISSN 1465-3842, pp. 25, November 2005. http://www.agentlink.org/newsletter/19/AL-19.pdf
- G. Di Marzo Serugendo, Marie-Pierre Gleizes, Anthony Karageorgos "AgentLink Second Technical Forum Group Self-Organisation in Multi-Agent", AgentLink Newsletter, Issue 17, ISSN 1465-3842, pp. 24-25, April 2005. http://www.agentlink.org/newsletter/17/AL-17.pdf
- G. Di Marzo Serugendo, Marie-Pierre Gleizes, Anthony Karageorgos "AgentLink First Technical Forum Group Self-Organisation in Multi-Agent Systems", AgentLink Newsletter, Issue 16, ISSN 1465-3842, pp. 23-24, December 2004. http://www.agentlink.org/newsletter/16/AL-16h.pdf

Papers

- Books

- Engineering Self-Organising Systems. S. Brueckner, S. Hassas, M. Jelasity, D. Yamins (Eds), LNAI 4335, Springer-Verlag, 2006.

- Engineering Self-Organising Systems. S. Brueckner, G. Di Marzo Serugendo, D. Hales, F. Zambonelli (Eds), LNAI 3910, Springer-Verlag, 2006.

- Engineering Self-Organising Systems - Methodologies and Applications. S. Brueckner, G. Di Marzo Serugendo, A. Karageorgos, R. Nagpal (Eds), LNAI 3464, Springer-Verlag, June 2005.

- Engineering Self-Organising Systems: Nature-Inspired Approaches to Software Engineering. G. Di Marzo Serugendo, A. Karageorgos, O. F. Rana, F. Zambonelli (Eds), LNAI 2977, Springer-Verlag, 2004.