Self-Organisation in Multi-Agent Systems

Marie-Pierre Gleizes[†] Gauthier Picard[‡]

[†] IRIT / Université de Toulouse
[‡] LSTI / École Nationale Supérieure des Mines de Saint-Étienne (ENSM.SE)

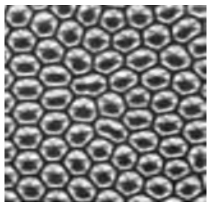
gleizes@irit.fr
picard@emse.fr

Contents

Contents

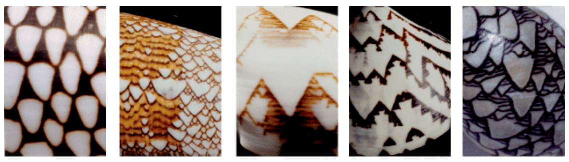
Non Living Systems

Bénard Convection Cells, Sand Dune Ripples



Living Systems

Giraffe, Rabbitfish, Zebra, Shells



Living Systems

Finger Prints, Morel

Social Systems
Ants, Wasps, Termites, Humans

Social Systems
Crustaceans, Ants

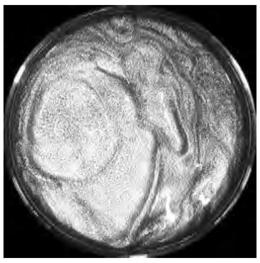
Social Systems
Fishes, Birds

Social Systems
Mammalians

Emergence

- In natural systems
  - Pattern formation
  - Behaviour
  - Phenomenon
- In artificial systems
  - Stable phenomena
  - Behaviour

Emergence in Natural Systems
Pattern Formation


Bénard Cells

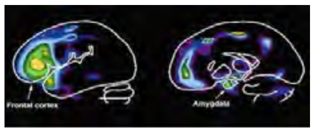
Belousov-Zhabotinsky Reaction

Emergence in Natural Systems
Behavior

Video: Guy Theraulaz, Laboratoire d'Ethologie et Cognition Animale, Toulouse France

Emergence in Natural Systems
Phenomenon

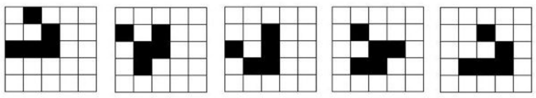
From simple neurons to the thinking brain [Searle, 1992]



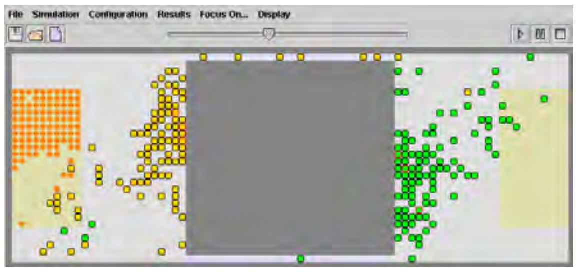
Emergence in Artificial Systems
Stable Phenomena

Conway's game of life [Gardner, 1970]

- Cellular automata [von Neumann, 1966]
- Emergence of a stable phenomenon: the glider
- A cell can be dead or alive
  - `If (dead and 3 neighbours alive) then alive`
  - `If (alive and 2 or 3 neighbours alive) then alive`
  - `Else dead`



Emergence in Natural Systems
Social Behavior



[Picard and Gleizes, 2005]

What is Self-Organisation in Artificial Systems?

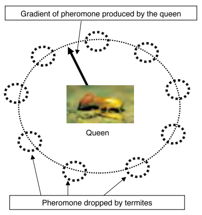
- Self-organisation is the mechanism or the process enabling a system to change its organisation without explicit external command during its execution time
  [Di Marzo-Serugendo et al., 2005]
- An autonomous transformation of the system topology (i.e. network connections) by its components as result of this network's functioning
  [Camps et al., 1998b]
- A set of dynamical interactions whereby structures appear at the global level of a system from interactions among its lower-level component... The rules specifying the interactions are executed on the basis of purely local information, without reference to the global pattern
  [Bonabeau et al., 1999]

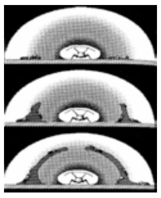
Strong and Weak Self-Organising Systems

- Strong self-organising systems are those systems where there is no explicit central control neither internal nor external
- Weak self-organising systems are those systems where, from an internal point of view, there is re-organisation maybe under an internal (central) control or planning

Strong and Weak Self-Organising Systems

Example


Gradient of pheromone produced by the queen
Queen
Pheromone dropped by termites



Termite nest construction : weak and strong self-organisation

Self-Organising Systems

[Farley and Clark, 1954]

- A system where a collection of interacting elements gives rise to patterns of behaviors that the individual elements are not capable of when they don't interact
- A system which changes its basic structure as a function of its experience and environment

➥ Emergent properties

- Absence of external control (autonomy)
- Decentralised control
- Dynamic operation (time evolution)
- Additional Properties
    - Fluctuations (noise/searches through options)
    - Symmetry breaking (loss of freedom/heterogeneity)

Characterisation

[Prigogine and Nicolis, 1977; Heylighen, 2001]

- Global order endogenous
- Emergence
- Simple local rules
- Instability (self-reinforcing choices/nonlinearity)
- Parameters sensitivity
- Multiple equilibria (many possible attractors)
- Criticality (threshold effects/phase changes)
- Redundancy (insensitivity to damage)
- Self-maintenance (repair/reproduction metabolisms)
- Adaptation (functionality/tracking of external variations)
- Complexity (multiple concurrent values or objectives)
- Hierarchies (multiple nested self-organised levels)

Requirements for Self-Organisation in MAS

- Two kinds of systems
    - System includes the environment: Ecosystem
    - System and environment can be differentiated: physical real environment
- Several agents
- Many interactions inside the system
- Limited perceptions
- Local behaviors at the agent level

Importance of the Environment

- Dynamic environment
- Coupling between the system and its environment
- At the macro level [Muller, 2004]
    - A collective (« unconscious ») memory
    - A global inscription medium
- At the micro level
    - The resources of the entities
    - An interaction medium
    - The coordination of interactions at various time scales (dissipation rate)
    - Constraints on the agent dynamics

What is Self-organisation in Natural Systems?

- A process in which pattern at the global level of a system emerges solely from numerous interactions among the lower level components of the system [Camazine et al., 2001]
- Rules specifying interactions among the system's components are executed using only local information without reference to the global pattern
- The pattern is an emergent property of the system, rather than a property imposed on the system by an external influence

What Does Emerge?

- The appearance of a property (or feature, or state) not originally observed as a functional characteristic of the system
  - Generally, higher level properties are regarded as emergent
- What can be qualified as emergent
  - Properties
  - Phenomena
  - Behaviour
  - Relevant/adequate function
  - State
  - ...

System Characteristics

- At least two levels (micro-macro)
- Dynamical
  - A form of self-maintained equilibrium
  - The ability to self-organise allowing an emergent phenomenon [Goldstein, 1999]
- Self-organisation, capacity of adaptation

Criteria to Decide whether there is Emergence

- Need to be observable at some level
- Novelty [Lewes, 1875; Van de Vijver, 1997]
- Coherence Irreducibility [Ali et al., 1997]
- Interdependency between levels [Langton, 1990]

$$local \underset{\leftarrow\ constraints}{\overset{causes\ \rightarrow}{\rightleftharpoons}} global$$

- Non linearity

The Role of the Observer

- Necessary to qualify the emergence
- Outside the system and no action on the system

Definition of Emergence

[Forrest, 1991; Muller, 2004]

- A phenomenon is emergent if and only if we have:
  - A system of interacting entities whose states and dynamics is expressed in a theory D
    - Example: the cells and its transition rules
  - The production of a phenomenon (a process, a stable state, an invariant) which is global relative to the former system:
    - Example: the regularities in the dynamics of the cells
  - The interpretation of the phenomenon via an inscription mechanism in another theory D':
    - Example: the glider and its laws
- The non-linearity of the interactions guarantees the irreducibility of D' to D

Towards an Operational Definition

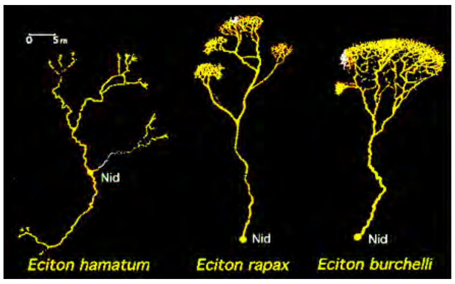
[Georgé, 2004; Georgé et al., 2004]

- The subject
  - A computational system has to realise a function which must be adequate to what is expecting a relevant user. This function, which may evolve during time, has to emerge
- The condition
  - This function is emergent if the coding of the system does not depend in any way of the knowledge of this function

  This coding has to contain the mechanisms allowing the adaptation of the system during its coupling with the environment, so as to tend anytime towards the adequate function

Emergence vs. Self-Organisation

- Emergent
  = Result of the collective
- Self-organisation
  = Means to obtain emergent phenomenon



Motivations

- Observations
  - Problems or applications too complex
  - Difficulty to have a complete global view, a global control
  - Self-organisation → adaptation capacity
  - Open systems
  - Incomplete specified problem
- Advantages
  - Simplification of the design: Bottom up approach
- Aims
  - Understand and control self-organisation
  - Find theories of the emergence

Emergence and Problem Solving

- Classical solving problem
  - Designer → Process leading to the solution
- Emergent solving problem
  - Designer → Agent, interaction and environment
  - The process by self-organisation builds the solution

Contents

Stigmergy

- « The work excites the worker » [Grassé, 1959]
  - → Behaviourist explanation indirect stimulus-responses
  - ← Observation on termites building behaviour
- Consequences
  - Direct interactions not necessary to coordinate the work of a group
  - Indirect interactions are sufficient
  - Indirect communication indirect between agents by the environment
- In social animals: termites, ants, bees, wasps, spiders, rats, etc.
  - Building behaviour
  - Recruitment
  - Division of labour
  - Prey transport
  - etc.

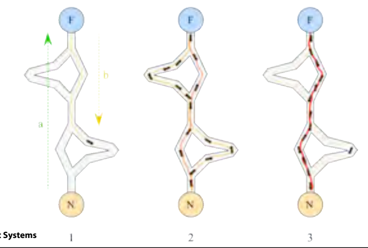
Stigmergy Requirements

- Stigmergy Elements
  - Environment
    - Central role
    - Dynamics
  - Individual interacting agents
    - Capabilities to move, perceive and act in the environment
    - Actions in the environment not for the others agents
- Stigmergy Design
  - Definition of the environment
    - What is perceived by agents
    - Which changes can be done by agents
    - What is the duration of the information: evaporation
  - Definition of the agents
    - How do they move
    - What they can do in the environment
    - In which state must they be to act: probalistic values

Ant Algorithms

[Dorigo et al., 1996]

- Probabilistic technique (metaheuristic)
  - Solving combinatorial problems
  - Finding good paths through graphs
- Stigmergic mechanism: pheromone trails
  - Deposited when food is found
  - Attracts ants (probabilistically)
  - ↓ Evaporates when no more used (bad source)
  - ↑ Reinforced when frequently used (good source



Ant Colony Optimization (ACO)

Arc Selection

$$p^k_{i,j}(t) = \begin{cases} \dfrac{\tau_{ij}(t)^\alpha \eta_{ij}^\beta}{\sum_{l \in J_i^k} \tau_{il}(t)^\alpha \eta_{il}^\beta} & \text{if } j \in J_i^k \\ O & \text{if } j \notin J_i^k \end{cases}$$

Pheromone Deposited

$$\Delta^k_{ij}(t) = \begin{cases} \dfrac{Q}{L^k(t)} & \text{if } (i,j) \in T^k(t) \\ O & \text{if } (i,j) \notin T^k(t) \end{cases}$$

Pheromone Update

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta^k_{ij}(t)$$

where:

- J_i^k, possible moves from i
- η_{ij}, visibility ($= 1/d_{ij}$)
- $\tau_{ij}(t)$, amount of pheromone on arc i,j
- α and β, parameters
- $T^k(t)$, visited arcs at time t
- $L^k(t)$, length of $T^k(t)$
- Q, parameter
- m, number of ants
- ρ, parameter

Social Spiders *(Anelosimus Eximius)*

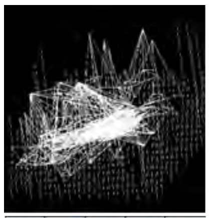
[Bourjot et al., 2003]

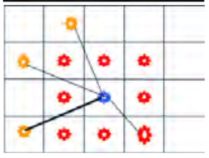
- Spiders are attracted by silk and by their other congeners
- Several individual spiders can succeed each other to build a web

Social Spiders

Modeling Issue

- Environment
  - Square grid composed of stakes with different heights
  - Initially without thread
  - Dynamical additions of spin threads
- Agents
  - Moving from one stake to another
  - Attraction by silk → contextual choice (probabilistic) of a given motion (function of the number of threads)
  - Putting silk at the top of a stake

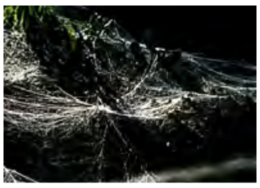




Social Spiders

System Dynamic

- Coordination by Stigmergy
  - Implicitly modelled in the behavior
  - Motion influenced by silk
  - More there is silk in a position, and greater is the chance to be chosen
  - No centralisation, no social reference
    - Dynamic relevant to individual and social spiders



Stigmergic Mechanisms

Multi-Agent Applications

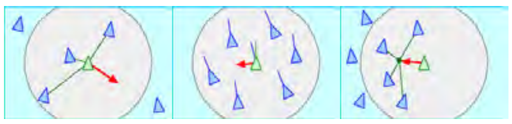
- Travelling salesman problem (TSP) [Dorigo et al., 1996]
- Computer network management, Ants foraging [Foukia and Hassas, 2004]
- Network routing, Ants foraging [Di Caro and Dorigo, 1998]
- Supply Network Management [Reitbauer et al., 2004]
- Coordination of unmanned vehicles [Parunak et al., 2002]
- Manufacturing control, Ants foraging [Brueckner, 2000; Armetta et al., 2004; Karuna et al., 2004]
- Security in networks, Ants foraging and immune system [Foukia, 2005]
- Mobile Ad-hoc NETworks [Brueckner and Parunak, 2004]

Flocking Behaviours

- Flock of birds, school of fish, or swarm of insects
- Realistic simulation of complex global behaviour with simple local behaviours
- First simulated in *Boids* [Reynolds, 1987]

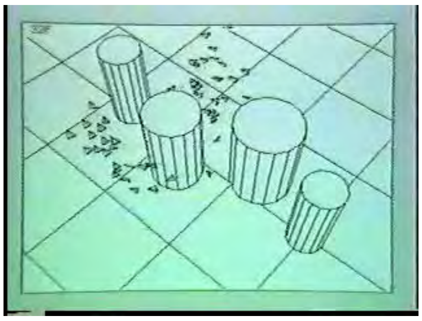
Flocking rules

Separation	avoid crowding neighbours
Alignment	steer towards average heading of neighbours
Cohesion	steer towards average position of neighbours



Flocking Behaviours

Boids Simulation (1986)



Reinforcement Mechanisms

- Principles
  - Consequences influence behaviour
  - Agent learns from experience and interactions
    - Role specialisation
    - Behaviour adaptation
  - Agent tries to maximise a reward
  - Children education, animal training
- Individual agents
  - Execute functions
  - Analyse the consequences of the execution of the function
- Rewards/Punishments
  - Associate them to the execution of functions
  - Local internal stimuli

Reinforcement Design

- Definition of local rules
  - Rewards reinforce a given behaviour
  - Punishments decrease a behaviour
- Definition of the agent
  - Recognize a given situation
  - Execute an action
  - Get some consequence
- Reinforcement Limitations
  - Difficulty to identify rewards and punishments
  - Necessity to control all sources of reinforcement
  - Difficulty to create internal changes
- Applications
  - Role based model [Weyns et al., 2004]
  - Task selection for foraging ants [De Wolf and Holvoet, 2003]
  - Role specialisation in a group of rats [Thomas et al., 2004]

Social Functions

- Human collective behaviour
  - Without central control, through self-organisation
- « Social Functions » emerge from human collective behaviour [Castelfranchi, 2001]
- Two kinds of social emergence
  - Emergent phenomenon is perceived by observer but has no effect on the society
  - Emergent phenomenon has an effect on the society
    - Self-producing + reinforcing social phenomenon
    - This actually is a « social function »
- Social functions = optimum order for society but...
  - Optimum order for society can be bad for individuals or for everybody
  - *Ex: prisons generate criminals that in turn feed prisons*

➥ Main application field: Multi-Agent Social Simulation

Trust-based Systems

- Human notion of trust
  - Uncertainty and partial knowledge
  - Human beings make choices, take decisions, learn by experience, adapt their behavior
  - Decisions implicitly rely on trust
    - Peers
    - Legal institutions
    - Business companies
- Idea
  - Human-like trust-based access control
  - To learn about peer behavior
  - To dynamically adapt access control policies

Trust-based Systems

Software Entities

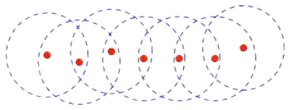
- Part of decentralised and distributed systems
- Autonomous, roaming
- Highly changing environment
  - Information changes and is not permanently valid
- Interactions occur locally
- Partial knowledge about the entities, and the environment
- Take decisions with local and incomplete knowledge
- Trust-based schema helps evaluating:
  - Good faith, correct functioning

➥ Main application fields: P2P, eMarket, Network Security

SECURE

[Cahill et al., 2003]

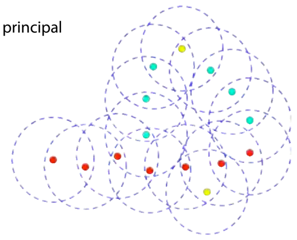
- Secure Environments for Collaboration among Ubiquitous Roaming Entities
- Goal: Trust-based access control
- Principals: interacting set of entities (human/computers, trusted or untrusted)
- Local trust values: Principals maintain local trust values about other principals
- Evidence
  - Direct observations: evaluated outcome of an interaction
  - Recommendations: asked or received (indirect observation)



SECURE
Scenario

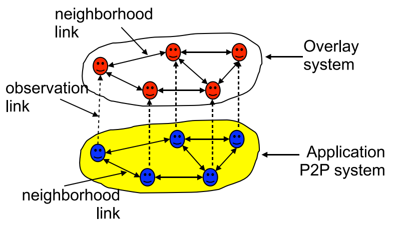
- Request of interaction
- Decision making process
  - Recognise principal
  - Evaluate trust value, evidence, risk implied by requested interaction
  - Application of Control Policy
- After interaction: trust value updated on the basis of evaluated outcome of the interaction
- Trust evolves with time

$\Longrightarrow$ allows to adapt behaviour of principal

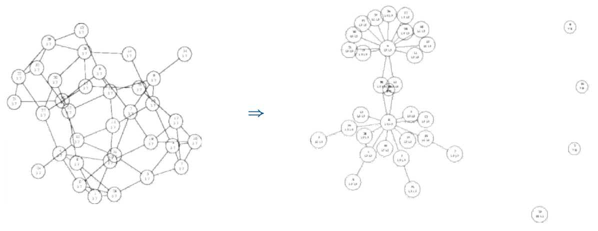


Reputation
[Grizard et al., 2006]

- Règles (R-normes) à respecter
  - Réputation objective des agents
  - Exclusion des agents malicieux
- Conventions (S-normes) locales
  - Réputation subjective
- Ré-organisation



Reputation
Example of Social Ordering



Gossip

General Definitions

- Light informal conversation for social occasions [World Reference Dictionary]
- Rumor or talk of a personal, sensational, or intimate nature [Webster Dictionary]

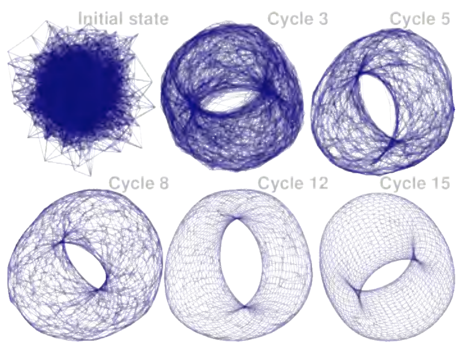
- Mechanism
  - Periodic exchange and update of information among members of a group
  - Allows: aggregation of global information inside a population, social learning
  - Parameters: neighborhood, level of precision of information
- Metaphor
  - Information spreading, knowledge exchange, group organisation, epidemics, virus spreading

➥ Main application fields: P2P protocols, sensor networks protocols

T-Man Algorithm
[Jelasity and Babaoglu, 2006]

- Generic protocol based on gossip communication model
- Goal: network topology management problem
- Principle
  - Nodes maintain local view of neighbours
  - Ranking function for reorganising the set of neighbours
    - Serves to reach the desired topology
  - Gossip message exchange
    - Choice of « closest » neighbour based on ranking function
    - Local exchange/combination of neighbours view
    - Nodes become closer and closer
  - Allows adaptation of neighbours list and re-organisation of the network topology
- Applications
  - Overlay networks supporting P2P systems
  - Sorting, Clustering, Distributed Hash table

T-Man Algorithm
Example: distance as ranking function



Coordination Spaces

- Linda [Gelernter, 1985]
  - Coordination model based on shared tuple spaces
  - Indirect communication
  - Insertion of tuples in the shared data space (out)
  - Retrieval of tuples from the shared data space (in or rd)
    - Retrieval is based on matching a given template
- Coordination spaces as middleware layers
  - Uncoupled interactions
  - Limited form of self-organisation
  - Decentralised control, anonymous and indirect interactions among agents

SwarmLinda

[Charles et al., 2004]

- Individuals = Active entities able to:
  - Observe their neighbourhood
  - Move in the environment
  - Change state of environment
- Environment
  - Context in which individuals work and observe
- State
  - Aspect of environment observed and changed by individuals
- SwarmLinda System = Network of nodes
  - Communicate with each other
  - Exchange tuples
- Principle: Ant Metaphor
  - Tuple Storage: Process performs an out request to a node N
    - Storage = ant sorting (Tuple-ant)
    - Tuple-ants carry tuple and drop tuple at specific nodes
  - Tuple Retrieval (in / rd)
    - Requests = ants looking for food (Template-ant)
    - Template-ants carry request and test at each node for matching tuples

SwarmLinda

Implementation

- Storage Implementation
  - Node N augments « scent » for that kind of tuple
  - If sufficient « scent » then keeps the tuple
  - If not looks for more suitable neighbour (highest concentration of that scent)
  - Tuple is sent to that node (that will do the same with the tuple: keep or send elsewhere)
- Retrieval Implementation
  - Node N determines if a local tuple matches the template
  - If no matching tuple: looks for highest scent for that kind of tuple in its neighbourhood
  - Request is sent to that node (that will do the same with the request)

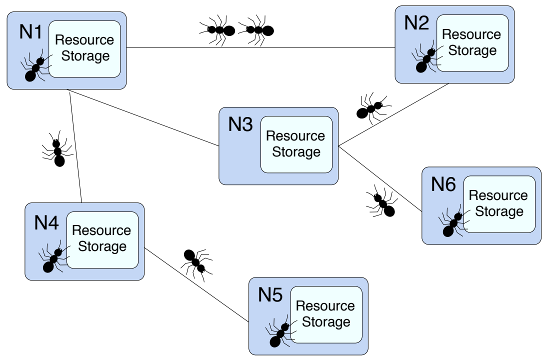
Anthill

[Babaoglu et al., 2002]

- Dynamic network of peer nodes
  - Adaptive agents travel through the network to solve complex problems
  - Interact with nodes
  - Cooperate with other agents
- P2P Applications
- Principle: Ant metaphor
  - Anthill system = Network of interconnected nests
  - Nest = peer entity sharing computational and storage resources
  - Nests handle requests of local users
    - Generate one or more ants roaming the network to satisfy the request
  - Ants
    - Observe environment
    - Perform local computation
    - Leave information on visited nests
    - Indirectly communicate with each other (stigmergy)
- Applications Development
  - Perform service requests to local node
  - Wait for replies

Anthill

Illustration



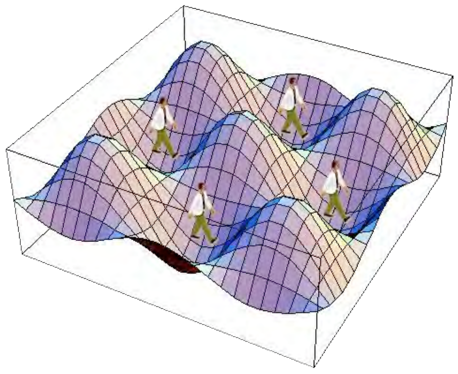
Co-Fields

Computational Fields [Mamei et al., 2004]

- Principle: Force Fields
  - Agents generate application-specific fields
  - Propagation of fields in environment according to field-specific laws
  - Composition of different fields
  - 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
- Application Development
  - Generation of fields, Definition of fields propagation, Agent reaction to fields
- Examples
  - Ants foraging, birds flocking

Co-Fields

Illustration



Co-Fields

Modelling of Ants Foraging

- 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
  - down to food or down to home
- Pheromone evaporation
  - Environment removes the wrinkle after elapsed time

TOTA

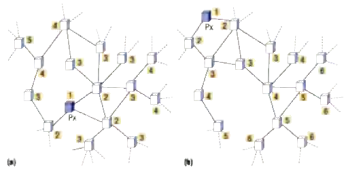
Tuples on the Air [Mamei and Zambonelli, 2005]

- Based on Coordination Space
- Uncoupled adaptive interactions
- Provides Context-awareness
- Follows Co-Field principle
- 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
- TOTA System
  - P2P Network of (Mobile) Nodes
  - Tuples injected in the system
    - Autonomously propagate and diffuse in the system
    - Propagation follows a specified pattern or propagation rule
- Application Development
  - Inject tuples (content+propagation rule)
  - Query local tuples (pattern-matching)

TOTA

Implementation

- Nodes maintain a limited list of neighbours
- TOTA-tuple = content + propagation rule
  - Content = information
  - Propagation rule
    - Describes how to diffuse the tuple
    - Scope (distance) of the tuple
    - Direction of propagation
    - Change of tuple content during propagation
- TOTA Middleware actively supports tuple propagation
  - If new node join the system, tuples are propagated to this new node (according to their propagation rules)



Contents

Problem Characteristics

Physically distributed	Ant foraging
Logically distributed	Experts cooperation
No known algorithm	Molecule conformation
Dynamic environment	Carrier robot
Open system	Internet market place

Classical System Design

- Global and top-down activity
  - To know the system finality
  - To know the interactions corpus in the future
- Arguments
  - To guaranty in a formal way that the real system realizes the « right » function
  - To optimize the treatment speeds and the very limited proprioceptive capacities of the first computers

➥ Relevance to represent and reason on the time, the space and the dynamic of the scalable world?

Difficulties for Designing

- Complex systems
  - No global control
  - Designer cannot control and build all the systems
- Dynamic of the environment
  - Adaptation
- Open systems
  - Adaptation
  - Robustness

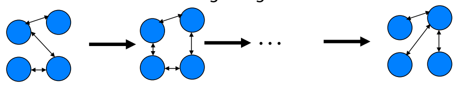
➥ Autonomy and adaptation are needed

Adaptive Systems

- Adapt their behaviour in order to react towards the environment dynamic
  - ← Achieve its task Improve their functioning
- Inspiration from natural systems
  - self-organisation social animals like ants, termite, . . .

Self-Organization in Artificial Systems

- The mechanism or the process enabling a system to change its organisation without explicit external control during its execution time [Di Marzo-Serugendo et al., 2005]
- Find a solution = find the right organisation



Solving process = succession of organizations
[Georgé et al., 2004; Picard and Glize, 2006]
- Problem Solving: Agents interact and evolve in a common environment
- Agents must have
  - Local rules
  - Local perceptions

Cooperation

General Definitions

- Cooperation is any group behavior that benefits the individuals more than if they were to act as independent agents
- More than two agents have to work together to achieve a task (resource, skill sharing) [Bernon et al., 2006]
- ...

- Two mains cooperative behaviours [Georgé et al., 2004]

  Anticipation: Try to act cooperatively and to avoid non cooperative acts
  Treatment: If an agent is in a non-cooperative situation, he acts to come back to a cooperative one

  ➥ Looks like exceptions in classical program
  ➥ Cooperation failure produced at the collective level but detected and treated at the local level

Cooperation

Multi-Agent System Context

- Definition of the environment
- Definition of the agents
- Definition of the cooperative attitude at the agent level
  - Cooperative interactions
  - Non cooperative interactions

Two levels of cooperation

- Between the environment and the multi-agent system
- Between agents

Adaptive Multi-Agent Systems Theory (AMAS)

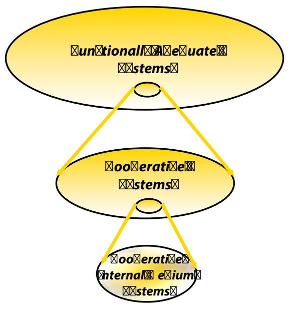
- Adaptive systems - Problem solving
  - *Find the right organisation between atoms in molecule by minimizing the global energy (agents = atoms)*
- Adequate function = what the system has to do to be « useful »
- Global function realized = result of the organizational process between agents
  - *The physical location of the atoms*
- Change the organization → change the global function
  - *If an atom moves the global energy can increase*
- To change the organization : self-organization by cooperation
  Autonomous parts + local rules
  Local criterion : cooperation

Adaptive Multi-Agent Systems Theory (AMAS)

Functional Adequacy Theorem [Camps et al., 1998a]

Theorem

For any functionally adequate system in a given environment, there is a system having a cooperative internal medium which realises an equivalent function



Adaptive Multi-Agent Systems Theory (AMAS)

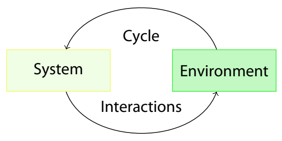
Hypothesis

- *S* system plunged into an environment
- *S* realizes a function f_S
- *S* composed of interacting agents
- Each agent realizes a partial function
- Interaction between agents can be
  - Cooperative
  - Antinomic
  - Indifferent
- **Organisation → Result**

➥ Cooperative Self-Organisation

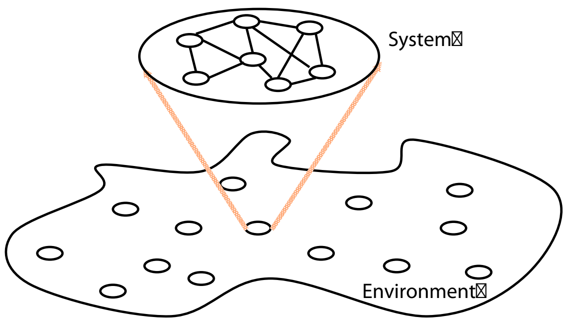
Adaptation by Coupling

[Maturana and Varela, 1994]



- Adaptation
  - In an autonomous manner
  - In relation with the medium
- Coupling
  - Mutual specification by interaction
  - In real time and not a priori
  - ➥ The system changes in run time its behaviour

Principle of Self-Organisation in AMAS

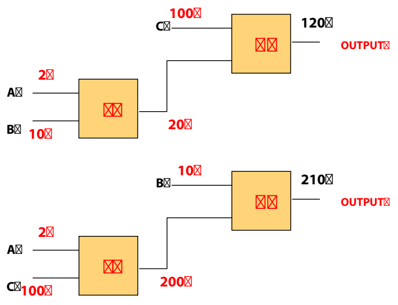


Time *t* : f_S

Cooperative Self-Organisation

Example: Emergent Programming [Georgé, 2004]

Simple example = 5 agents (+ , ∗, 3 constants)

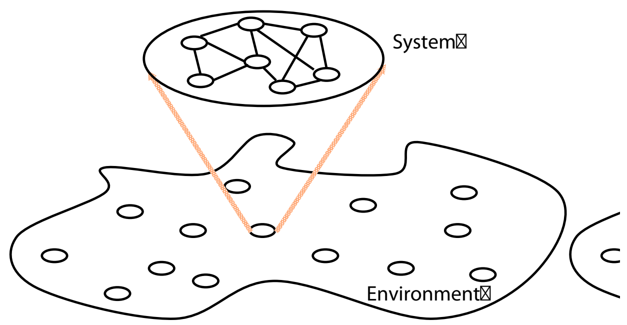


AMAS Theory: Emergence

- Artificial Systems

Object:	The global function of the system emerges
Condition:	the coding of the system is not explicitly ordered by the knowledge of this global function [Georgé et al., 2004]
Method:	**self-organisation by cooperation**

Principle of Self-Organisation in AMAS



Time $t : f_S$ Perception Time $t+1 : f_S^* +$ Action

Cooperative Self-Organisation

Cooperation = Engine for Self-Organisation

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

Definition: Cooperation in AMAS

An agent is cooperative if:

c_{per} All perceived signals must be understood without ambiguity

c_{dec} The received information is useful for the agent's reasoning

c_{act} Reasoning leads to useful actions towards others agents

➥ Proscriptive appraoach: agents must avoid or resolve non cooperative situations (NCS): $\neg c_{per}$ or $\neg c_{dec}$ or $\neg c_{act}$

Non Cooperative Situations (NCS)

- **Anticipation**: try to avoid "problems"
- **Exception treatments:** "detection and handler execution"
- An agent must have a cooperative attitude
  - It detects and repairs *Non Cooperative Situations*
  - It tries to avoid Non Cooperative Situations which can be anticipated by itself
  - It always tries to be cooperative BUT an agent is benevolent and not altruistic → sometimes Non Cooperative Situations occur

Non Cooperative Situations can be viewed as **exceptions** at the agent's interaction level

Non Cooperative Situations (NCS) (cont.)

Definition of a cooperative situation from the local point of view of an agent

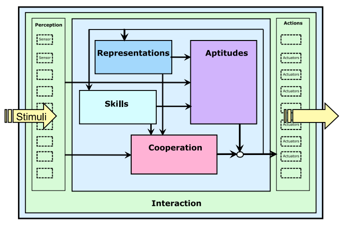
- All perceived signals must be understood without ambiguity
  - Incomprehension
  - Ambiguity
- The received information is useful for the agent's reasoning
  - Unproductiveness
  - Incompetence
- Reasoning leads to useful actions towards others
  - Conflicts
  - Concurrency
  - Uselessness

Cooperative Agent Architecture

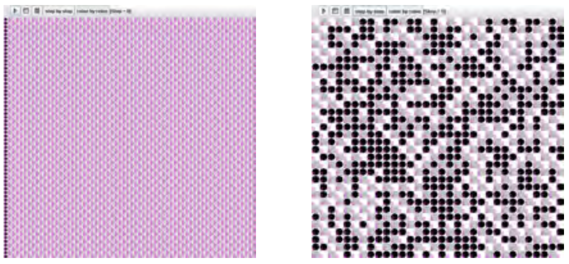
- Is autonomous
- Respects the criteria of locality
- Ignores the global function of the system

Fundamental activities : perceive, decide and act in the world

- a cooperative situation → realises its function
- an uncooperative situation (failure) → acts to come back in a cooperative state



Example 1: *n*-Queens Problem



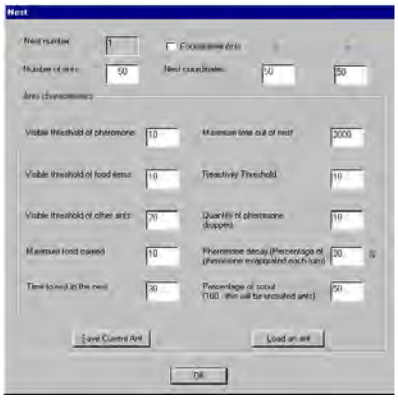
[Picard and Glize, 2006]

Example 1: *n*-Queens Problem

NCS Examples

1. Imcomprehension ($\neg c_{per}$)?
2. Ambiguity ($\neg c_{per}$)?
3. Incompetence ($\neg c_{dec}$)?
4. Unproductiveness ($\neg c_{dec}$)?
5. Concurrency ($\neg c_{act}$)? Two queens on a same cell
   - *Condition:* Be on the same cell than another queen
   - *Action:* Move to the less constrained visible cell
6. Conflict ($\neg c_{act}$)? Two queens can attack the same cell
   - *Condition:* Be on a cell that others can attack
   - *Action:* Move with minimum impact on others' constraints violation OR let the others performing a movement
     (by analysing the less constrained cell and the most constrained queen that perceives it)
7. Uselessness ($\neg c_{act}$)? A queen sees a less constrained cell
   - *Condition:* See a less constrained cell
   - *Action:* Move to the less constrained cell

Example 2: Foraging Ants



Nest

Nest number	1	☐ Foodsearcher Ants		
Number of ants	50	Nest coordinates	50	50

Ants characteristics

Visible threshold of pheromone	10	Maximum time out of nest	2000
Visible threshold of food items	10	Reactivity Threshold	10
Visible threshold of other ants	20	Quantity of pheromone dropped	10
Maximum food carried	10	Pheromone decay (Percentage of pheromone evaporated each turn)	30 %
Time to rest in the nest	30	Percentage of scout (100 - this will be recruited ants)	50

Example 2: Foraging Ants

Cooperative Behaviour of an Robot-Ant

- Cooperative social attitude
- Dynamic environment *rightarrow* a lot of non cooperative situations
- Behaviour : come back to cooperative interactions
- ➥ Specify all non cooperative situations

Example 2: Foraging Ants

Concurrency Situation 1

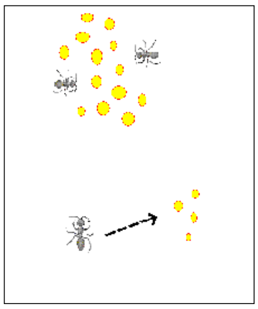
Two choices:

1. Follow the pheromone track
2. Go towards new foods

➥ To avoid concurrency, the robot-ant go towards new food location even if the pheromone is in a big quantity

Example 2: Foraging Ants

Concurrency Situation 2



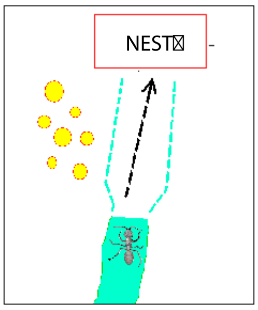
Two choices:

1. Go towards new foods already found by others ants
2. Go towards new foods unused

➥ To avoid concurrency, the robot-ant go towards the unexploited food location even if there is less food than at the other location

Example 2: Foraging Ants

Cooperative Attitude



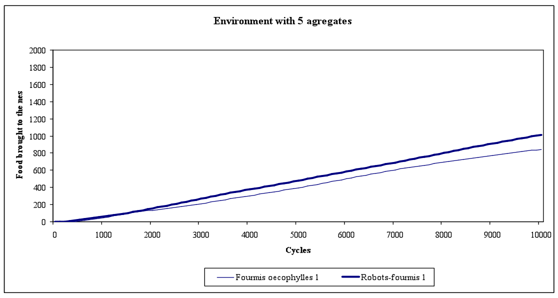
Two choices:

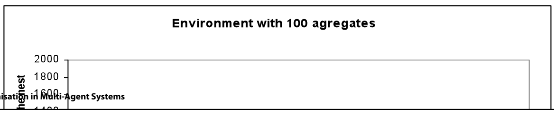
1. Go directly towards the nest
2. Go directly towards the nest in dropping pheromone

➥ It is a spontaneous communication: the robot-ant drops more pheromone when coming back

Example 2: Foraging Ants

Comparison with "real" ants (œcophylles Ants)





AMAS Theory: Design of Adaptive MAS

- The approach to design adaptive systems is a **bottom-up** approach
- Designer has to:
  - Determine agents Find for all types of agents, all **generic** Non Cooperative Situations an agent can encounter by using some meta-rules: ambiguity, incomprehension, conflict, concurrence, unproductiveness, incompetence, uselessness
  - For every Non Cooperative Situation, provide a handler to treat this NCS
- In general the treatment leads to change the interactions between agents

➥ Change the organisation

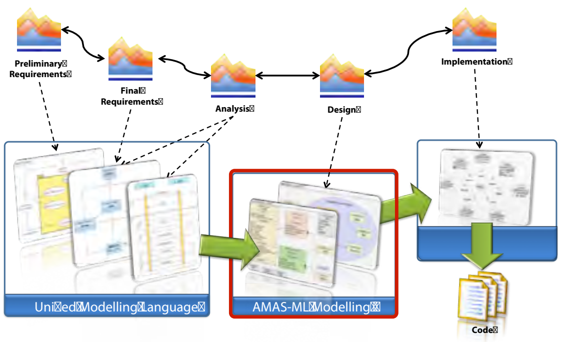
➥ A method ADELFE [Picard and Gleizes, 2004]

ADELFE

- Aims
  - Not general agent-based methodology
  - ➥ Exploiting the AMAS Theory → cooperation
  - ➥ Systems open, adaptive to change in the environment
  - For engineers aware of MAS
- Principles
  - Not start from scratch → RUP and Notation, based on (A-) UML
  - Top down approach: analysis phase - identification of the agents
  - Bottom up approach: design phase – agent design
- Keypoints
  - Environment characterisation
    Definition of NCS between the system and the environment
  - Non cooperative situations are exceptions at the agent interaction level
  - Verification AMAS adequacy
  - Agent Identification
  - Agent Design → model - stereotypes
    Guide to define local rules for agent behaviour : cooperative attitude

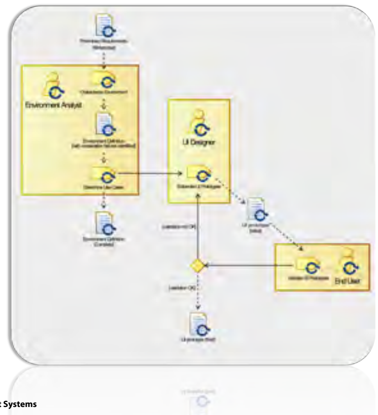
ADELFE

Process

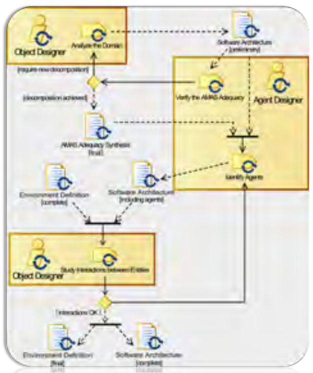


ADELFE

Requirements

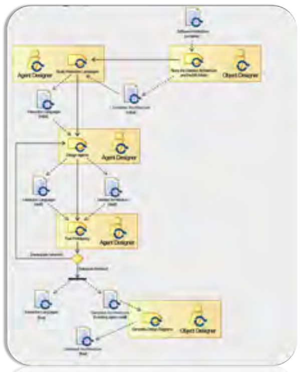


ADELFE
Analysis

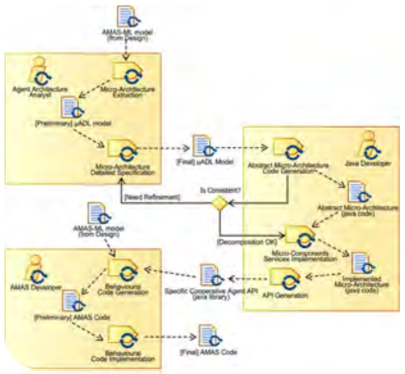


Object Designer
Analyze the Domain
Software Architecture [preliminary]
[require new decomposition]
Verify the AMAS Adequacy
Agent Designer
[decomposition achieved]
AMAS Adequacy Synthesis [final]
Identify Agents
Environment Definition [complete]
Software Architecture [including agents]
Object Designer
Study Interactions between Entities
[interactions OK]
Environment Definition [final]
Software Architecture [complete]

ADELFE
Design



ADELFE
Implementation

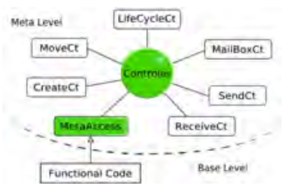


AMAS-ML model (from Design)
Agent Architecture Analyst
Micro-Architecture Extraction
[Preliminary] µADL model
Micro-Architecture Detailed Specification
[Final] µADL Model
Abstract Micro-Architecture Code Generation
Java Developer
Is Consistent?
[Need Refinement]
Abstract Micro-Architecture (java code)
AMAS-ML model (from Design)
[Decomposition OK]
Micro-Components Services Implementation
Implemented Micro-Architecture (java code)
AMAS Developer
Behavioural Code Generation
Specific Cooperative Agent API (java library)
API Generation
[Preliminary] AMAS Code
Behavioural Code Implementation
[Final] AMAS Code

ADELFE
Functional/Operational Adaptation

- Operational level
  - Micro-architecture with components
  - Abilities of the agents
- Functional (behavioural) level
  - Main function
  - Use services from the operational level

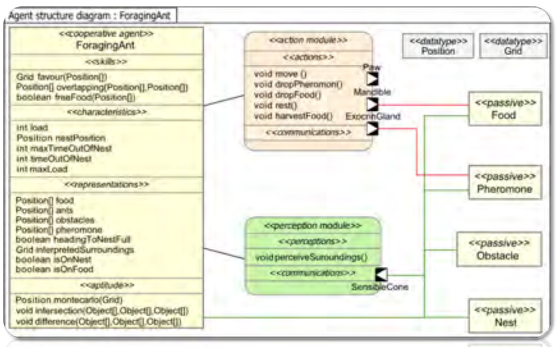
➥ Operational Adaptation → **Replace a micro-component**



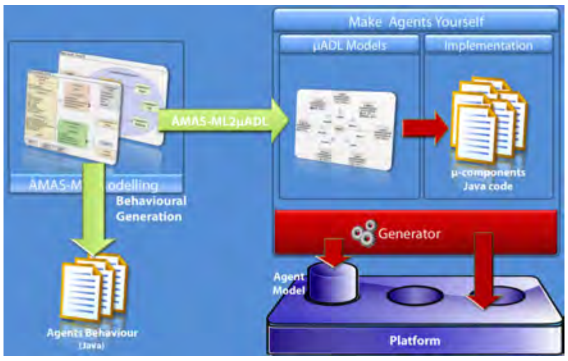
ADELFE
AMAS-ML (AMAS-Modeling Language)

- Characteristics
  - Language dedicated to AMAS
  - Metemodel defined in Ecore (Eclipse)
  - 4 packages:
    - Core: basic elements
    - System: the system inside its environment
    - Agent: coopertaive agent model
    - Cooperation: decision elements (rules)
- Strength of the Metamodel
  - Formalisation
    - Agent model
    - Cooperation expressed with rules
  - Evolutive approach for tools definition
    - Graphical editor, text
    - Model transformation
    - Code generation

ADELFE
Example of Model



ADELFE
Transformation chain



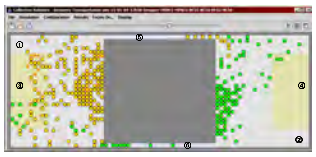
Contents

Cooperation
Multi-Agent Applications

- AMAS applications
  Main works: `http://www.irit.fr/SMAC`
  - Flood forecast (generic river behavior model building without geo-physical knowledge)
  - Optimized ant simulation
  - Robot transportation
  - Adaptive ontology / semantics building
  - Autonomous mechanical design
  - Emergent Programming (instruction agents)
  - Manufacturing control
  - Time tabling
  - Bioinformatics
- British Telecom [Marrow et al., 2003]

Collective Robotics
Navigation in Constrained Environments

- Robots
  - Autonomous
  - Resource transportation task
  - Micro-level entities



- World
  - Two rooms
  - Narrow corridors separate the rooms

➥ Spatial interferences

Modules
Interactions

- Perceptions
  - Limited perception cone
  - Proximity sensors
  - Identification of seen object type
  - Global position (ex: GPS)
  - No direct communication
- Actions
  - *rest*, *pick*, *drop*, *forward*, *backward*, *left* and *right*
  - Robots cannot drop boxes anywhere
  - No communicative acts

Modules
Knowledge

- Skills
  - Knowledge about the task to perform (goal)
  - Preferences on the next action to reach the current goal : *reach claim zone* ($goal_1$) and *reach laying zone* ($goal_2$)
  - Intrinsic characteristics : speed, carried box, reflex preferences
- Representations
  - Limited knowledge about the environment and itself
  - Limited memory : past position, direction, goal and action

Modules
Decisions

- Aptitudes
  - Enables an agent to choose an action in terms of its perceptions, skills and representations
  - Chooses among possible actions what will be the next action to reach the goal
- Cooperation
  - Enables an agent to choose an action in terms of its perceptions, skills and representations
  - Chooses among possible actions what will be the next action to be cooperative
  - Cooperative behaviour subsumes nominal one

Action Choosing
Implementing Nominal Behaviour

At each time t, a robot has to choose between different actions that are proposed by the two decision modules (aptitudes and cooperation)

Nominal behaviour

At time t, each action act_j of the robot r_i is evaluated :

$$V_{r_i}^{nomi}(act_j,t) = wp_{r_i}(act_j,t) + wm_{r_i}(act_j,t) + wr_{r_i}(act_j)$$

with:

- $V_{r_i}^{nomi}(act_j,t)$: value for the action act_j at time t for the robot r_i
- $wp_{r_i}(act_j,t)$: calculated value in terms of perceptions
- $wm_{r_i}(act_j,t)$: calculated value in terms of memory
- $wr_{r_i}(act_j,t)$: calculated value in terms of reflexes

Action Choosing
Implementing Cooperative Behaviour

Cooperative behaviour

As for aptitudes, an action preference vector is generated by the Cooperation Module:

$$V_{r_i}^{coop}(act_j,t) = wp'_{r_i}(act_j,t) + wm'_{r_i}(act_j,t) + wr'_{r_i}(act_j)$$

Agent's behaviour

$$V_{r_i}(t) = V_{r_i}^{nomi}(t) \lessdot V_{r_i}^{coop}(t)$$

Cooperative Unblocking
Reactive Cooperation

- Observation
  - Beyond two robots, the nominal behavior cannot be adequate : spatial interferences

  ex: If two robots, a first one carrying a box and moving to the laying zone and a second one moving to the claim zone to pick a box, meet in a corridor, the circulation is blocked
  - It is necessary to provide cooperative behaviors to robots

Blocking NCS (in the Cooperation Module)

- *A robot is blocked*

  ➥ "Move by side" or "The robot closer to its goal has priority"
- *A robot is returning*

  ➥ "Move by side" or "Continue moving ahead until blocked"
- Conflict or Uselessness NCS

Cooperative Unblocking
Example : *a robot is returning*

Condition	Action
$ret \wedge freeR$	$\nearrow V_{r_i}^{coop}(t,right)$
$ret \wedge freeL$	$\nearrow V_{r_i}^{coop}(t,left)$
$ret \wedge \neg(freeL \vee freeR) \wedge ant \wedge toGoal \wedge cGoal$	$\nearrow V_{r_i}^{coop}(t,backward)$
$ret \wedge \neg(freeL \vee freeR) \wedge ant \wedge toGoal \wedge \neg cGoal$	$\nearrow V_{r_i}^{coop}(t,forward)$
$ret \wedge \neg(freeL \vee freeR) \wedge ant \wedge \neg toGoal$	$\nearrow V_{r_i}^{coop}(t,backward)$
$ret \wedge \neg(freeL \vee freeR) \wedge \neg ant$	$\nearrow V_{r_i}^{coop}(t,forward)$

- *ret*: r_i is returning
- *freeR*: right cell is free
- *freeL*: left cell is free
- *ant*: in front of an antinomic robot
- *toGoal*: r_i is moving to goal
- *cGoal*: r_i is closer to its goal than its opposite one
- $\nearrow$: increasing

Cooperative Anticipation

- Observation
  - Unblocking mechanisms are not sufficient : robots repeat their errors
  - As an optimization, it is possible to provide cooperative anticipative behaviour and memory

Anticipation (in the Cooperation Module)

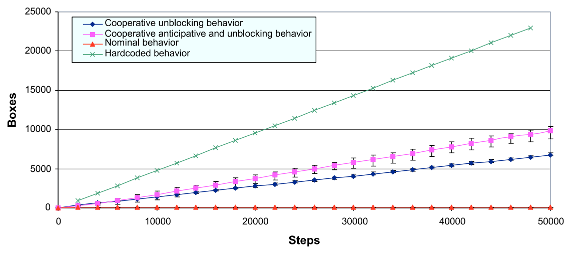
- To avoid "risky" areas
- *A robot sees an antinomic robot*

  ➥ "Move by side" or "Move forward"

Memory (in the Representation Module)

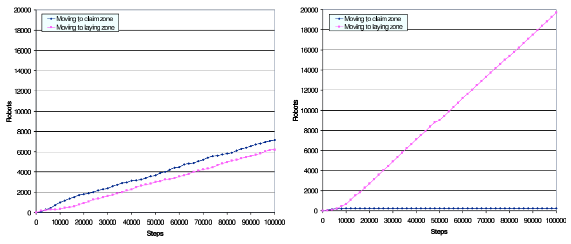
- Using virtual markers : $\langle posX(r_j,t), posY(r_j,t), goal(r_i,t), w\rangle$
- Marker with w and situated in the direction dir at a distance d induces that $V_{r_i}^{coop}(t, dir_{opp})$ increases of w

Reaction vs. Anticipation



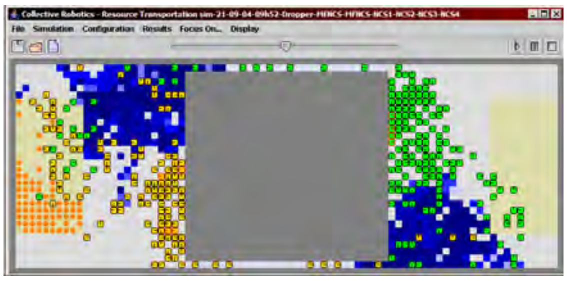
Number of transported boxes for 15 simulations (300 robots, 2 corridors, 5-ranged perception)

Emergence



Number of incoming robots for a corridor and for the two cooperative behaviors: unblocking behavior (left) and anticipation unblocking behavior (right)

Emergence

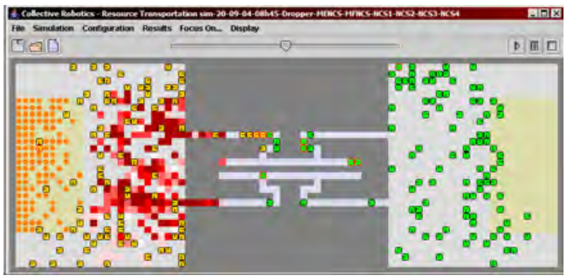


Collective Robotics - Resource Transportation sim-21-09-04-09h52-Dropper-MENCS-MFNCS-NCS1-NCS2-NCS3-NCS4
File
Simulation
Configuration
Results
Focus On...
Display

Positioning of all the virtual markers (dark squares) for all the robots and the two goal

Emergence

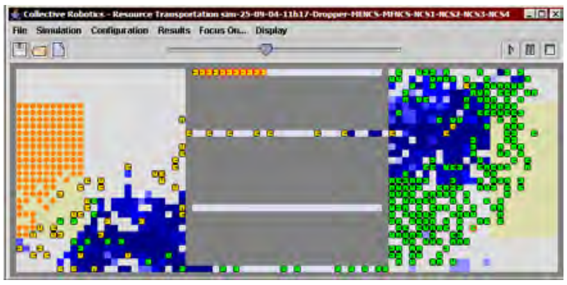
Robustness in Difficult Environments



Collective Robotics - Resource Transportation sim-20-09-04-08h45-Dropper-MENCS-MFNCS-NCS1-NCS2-NCS3-NCS4
File
Simulation
Configuration
Results
Focus On...
Display

Positioning of all the virtual markers (dark squares) for the goal *reach laying zone* in a difficult environment (with deadends)

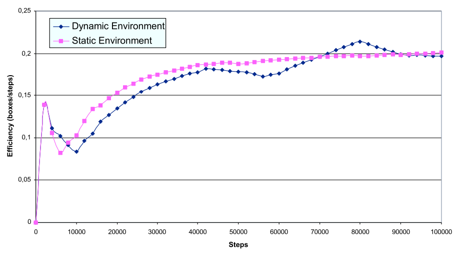
Adaptation to Dynamics



Collective Robotics - Resource Transportation sim-25-09-04-11h17-Dropper-MENCS-MFNCS-NCS1-NCS2-NCS3-NCS4
File
Simulation
Configuration
Results
Focus On...
Display

Positioning of the virtual markers (dark squares) in a dynamic environment with two closed corridors

Adaptation to Dynamics



Compared efficiency between a simulation in a static environment with two corridors and another one in an dynamic environment with 4 corridors which 2 corridors are randomly closed every 10,000 steps

Analysis

- Application of a cooperative agent model to a multi-robot transportation problem
- Emergence of a global behaviour : robots dedicates corridors without manipulating "corridor" notion
- Pros & Cons
  - ✓ No communication
  - ✓ No shared memory
  - ✓ Robust behaviour (number of corridors, corridor closure, etc.)
  - ✓ No global feedback
  - ✗ Absolute position
  - ✗ Parameters are difficult to set

Frequency Assignment

MAS Description

- Agent
  - An agent is responsible for assigning a value to a path
  - Main characteristics: value, difficulty
  - Local view
- Environment
  - Social: neighboring agents, sharing constraints
  - Physical: domains, constraints

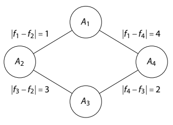
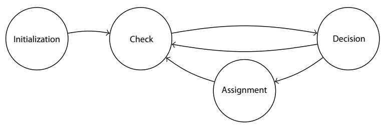


Figure: Simple example with 4 agents and 4 hard constraints

Agents' Behavior



- Decision
  - Each agent compares its difficulty with its neighbors
  - The agent with the highest difficulty is elected
- Assignment
  - Only concerns elected agents
  - Assignment is the only possible action
  - The best assignment, from the agent's viewpoint, is chosen

Communication

- Description
  - Using direct messages / asynchronous
  - Exchanging values and difficulties between neighbors
- Assignment Session
  - Blocks the neighbors of the elected agent (synchronization point)
  - The elected agent invites the neighbors, and waits until all neighbors acknowledge
  - Updates the elected agent's views to choose the best action
  - Neighbors can *cancel* a session
  - Solves conflicts during decision
- Ineligible Agents
  - 2 consecutive elections (agent-level tabu)
  - all constraints satisfied

Difficulty Measurement

- Expresses the distance to a good solution, from the agent's viewpoint
- Empirically determined (heuristic)
- Sorted tuple of sub-criteria

$$d_x^y = [Im_x^y, Po_x^y, NS_x^y, Ol_x^y]$$

Im Improvement, value domains analysis

$$\forall x \in A, Im_x = NS_x - min\{NS_x(f,p),\ (f,p) \in F(x) \times P(x) \setminus (f,p)_x\}$$

Po Possibilities, constraint difficulty

$$\forall x \in A, Po_x = min\{|FPS(c^x)|,\ c^x \in Cl_x \text{ and } c^x = false\}$$

NS Number of unsatisfied constraints

Ol Number of assignments within the neighborhood since the last constraint satisfaction

$$\forall x \in A,\ Ol_x = max\{\ Tl_x(c^x,\ c^x \in Cl_x|)\}$$

Choosing Values

1st reduction of the value domain: discriminant criterion

- during the comparison of two agents' difficulties, one criterion discriminates
- the choice of values is based on this discriminant criterion

Im	: values that maximize the improvement
Po	: set of constraints with the minimum of possibilities
NS	: set of constraints with the maximum of possibilities
Ol	: the oldest constraints
Eq	: the constraints shared with the agents having a difficulty equal to the agent's difficulty

Choosing Values (cont.)

2nd reduction of the value domain: dialogging with neighbors

- an assignment can improve *immediately* the situation of the neighbor
- an assignment can improve *possibly* the situation of the neighbor

➡ utility of a value

$$u_x^y(f_x,p_x) = max\{|S_y(f_y,p_y)|,\ (f_y,p_y) \in F_y \times P_y \text{ and } (f,p)_x^y = (f_x,p_x)\}$$

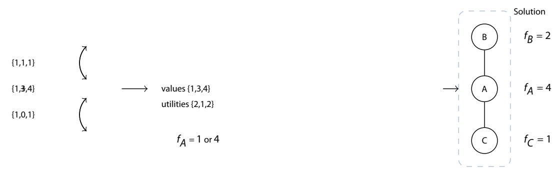
- choosing the value that maximizes the sum of utilities

3rd reduction of the value domain: the elected agent's choice

- "selfish" choice
- depending on the immediate improvement for each value
- in case of equality: randomize

Example

- 3 paths: A, B et C
- $D_f = \{1,2,3,4\}$
- $|f_A - f_B| = 2$
- $|f_A - f_C| = 3$



Convergence

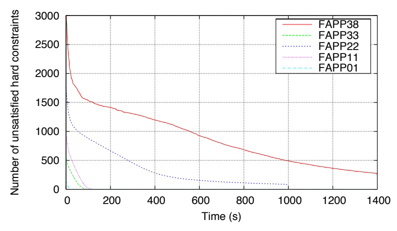


Figure: Convergence to solutions

✓ global decreasing of the number of unsatisfied constraints
✗ the solving process is slow

Assignments

Table: Percentage of unsatisfied CI

FAPP Instance	01	11	22	33	38
CI	168	978	1799	578	3112
% Unsatisfied	0	0,015	3,49	0,004	6,82

Table: Assignments

FAPP Instance	01	11	22	33	38
Assign.	107.24	547.02	964.81	338.35	1146.18
Assign./CI	0.64	0.56	0.54	0.59	0.37
Assign./Time(s)	17.87	3.42	0.96	2.26	0.82
σ	5.33%	6.89%	5.36%	2.28%	3.88%

✗ the solving process is slow: few assignments per second
✓ assignments are well-chosen
✓ independence from initial state

Message Traffic

Table: Message traffic

FAPP Instance	01	11	22	33	38
Agents	200	1,000	1,750	650	2,500
Total Messages	2,335	15,841	33,247	8,348	48,727
Messages/Agent	11.7	15.8	19	12.8	19.5

Table: Percentage of canceling messages

FAPP Instance	01	11	22	33	38
% canceling	4.21	6.02	13.56	5.22	10.28

Decision Criteria Relevance

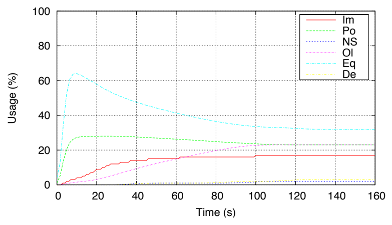


Figure: Distribution of the discriminant criteria for the instance FAPP11

Discussion

- **Communication**
  - ✓ Information updates can be limited
  - ✓ Assignment sessions can be limited
  - Limiting the invitation time
  - Limiting the number of invited agents: preferences
  - Agents may be invited to more than one session
- **Difficulty Evaluation**
  - ✓ criteria are generic
  - sorting criteria is an open question
- **Soft Constraints and Optimization**
  - ✗ CEM are more numerous
  - Two kinds of neighborhoods can be considered
  - How to include CEM in difficulty measurement ?
- **Comparison to Other Approaches**
  - ✓ No addition of virtual neighborhood (APO, ABT)
  - ✓ No hierarchy (ADOPT) → openness
  - Termination criterion is time limit
  - ✗ 2 times less efficient than ADOPT on graph-coloring (but different topology)

Aeronautical Design

- Tools to ease complex systems design
- Interaction with the designer
- Need to adapt to changes
- Continuous domains
- Examples
  - Preliminary Design
    - Agents = disciplines (geometry, aerodynamics, mass, etc.)
    - Cooperation = negotiation of function parameters
    - Objective = optimisation of objective functions under constraints
  - Mechanical Design
    - Agents = components (bars, edges, etc.)
    - Cooperation = négotiation of dimensions and properties between neighboring components
    - Objectif = follow a predefined trajectory

Preliminary Design

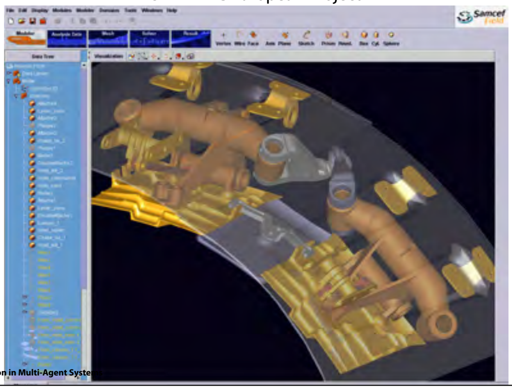
[Welcomme et al., 2007]

ID4CS Project

Mechanical Design

[Capera et al., 2004]

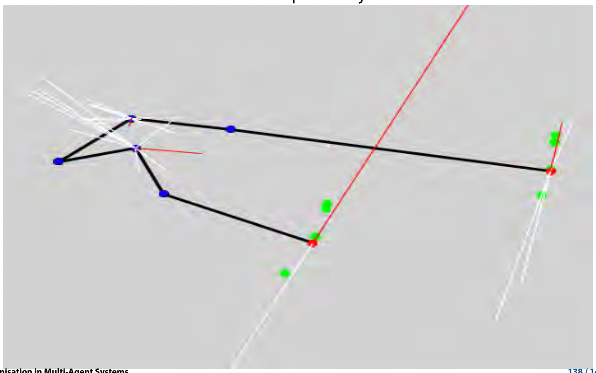
SYNAMEC European Project



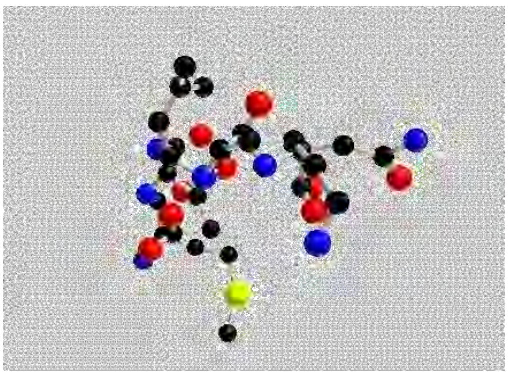
Mechanical Design

[Capera et al., 2004]

SYNAMEC European Project



Molecular Conformation



Contents

Conclusion

- Multi-disciplinary domain
  - Natural sciences
  - Social sciences
  - Systemics
  - ...
- Bottom-up approach
  - Model/design environment (interfaces)
  - Design agents
  - Design self-organisation rules
- Main advantages
  - ✓ Adaptation to dynamics
  - ✓ Problems tracing
  - ✓ Anytime

Conclusion (cont.)

Main scientific issues

- Engineering
  - ✗ No control/emergence → No proof of convergence, stability, etc.
    - Needs of formal/semi-formal/experimental proofs
  - How to choose a mechanism for a given problem ?
- Find new mechanisms
- From weak to strong emergence: organisational loop
  1- self-organisation
  2- organisational reflection and memorisation
  3- perturbation → self-organisation (no organisational reflexion) or re-organisation (organisation reflexion)
  ...

References

K. Aberer. Distributed Data Management: Part 3 - Peer-2-Peer Systems. Course, EPFL, 2004.

S.M. Ali, R.M. Zimmer, and C.M. Elstob. The Question Concerning Emergence : Implication for Artificiality. In Dubois, D.M., editor, *First International Conference on Computing Anticipatory Systems (CASYS'97)*, 1997.

F. Armetta, S. Hassas, S. Pimont, and E. Gonon. Managing Dynamic Flows in Production Chains Through Self-Organization. In *Engineering Self-Organising Systems: Methodologies and Applications*, volume 3464 of *Lecture Notes in Computer Science (LNCS)*, pages 240--255. Springer, 2004.

O. Babaoglu, H. Meling, and A. Montresor. Anthill: A Framework for the Development of Agent-Based Peer-to-Peer Systems. In *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02)*, pages 15--22. IEEE Computer Society, 2002.

U. Bellur and N. C. Narendra. Towards a programming model and middleware architecture for self-configuring systems. In *Proceedings of the First International Conference on COMmunication System softWAre and MiddlewaRE (COMSWARE 2006), Jamuary 8-12, 2006, New Delhi, India*. IEEE CS, 2006.

C. Bernon, V. Chevrier, V. Hilaire, and P. Marrow. Applications of Self-Organising Multi-Agents Systems: An Initial Framework of Comparison . *Informatica*, 30(1):73--82, 2006.

E. Bonabeau, M. Dorigo, and G. Theraulaz. *"Swarm Intelligence: From Natural to Artificial Systems"*. Oxford University Press, 1999.

C. Bourjot, V. Chevrier, and V. Thomas. A New Swarm Mechanism based on Social Spiders Colonies : from Web Weaving to Region Detection. *Web Intelligence and Agent Systems: An International Journal (WIAS)*, 1(1):47--64, 2003.

S. Brueckner. *Return from the Ant: Synthetic Ecosystems for Manufacturing Control*. PhD thesis, Department of Computer Science, Humboldt University Berlin, 2000.

S. Brueckner and H. V. D. Parunak. Self-Organizing MANET Management. In *Engineering Self-Organising Systems, Nature-Inspired Approaches to Software Engineering [revised and extended papers presented at the Engineering Self-Organising Applications Workshop, ESOA 2003, held at AAMAS 2003 in Melbourne, Australia, in July 2003 and selected invited papers from leading researchers in self-organisation]*, volume 2977 of *Lecture Notes in Computer Science (LNCS)*, pages 20--35. Springer, 2004.

V. Cahill, B. Shand, E. Gray, C. Bryce, and N. Dimmock. Using trust for secure collaboration in uncertain environments. *IEEE Pervasive Computing Magazine, special issue Dealing with Uncertainty*, 2(3):52--61, 2003.

S. Camazine, J.-L. Deneubourg, N. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-Organization in Biological Systems*. Princeton University Press, 2001.

V. Camps, M-P. Gleizes, and P. Glize. Une théorie des phénomènes globaux fondée sur des interactions locales. In J-P. Barthès, V. Chevrier, and C. Brassac, editors, *Systèmes multi-agents -- de l'interaction à la socialité -- Actes des 6èmes JFIADSMA*, pages 207--220. Editions Hermès, 1998a.

References (cont.)

V. Camps, M.-P. Gleizes, and S. Trouilhet. Properties Analysis of a Learning Algorithm for Adaptive Systems. *Journal of Computing Anticipatory Systems*, 1:223--233, 1998b.

D. Capera, M.-P. Gleizes, and P. Glize. Mechanism Type Synthesis based on Self-Assembling Agents. *Journal of Applied Artificial Intelligence*, 18(9-10): 921--936, 2004.

S. Capkun, L. Buttyán, and J.-P. Hubaux. Self-Organized Public-Key Management for Mobile Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 2(1):52--64, 2003.

C. Castelfranchi. The theory of social functions: challenges for computational social science and multi-agent learning. *Journal of Cognitive Systems Research*, 2(1):5--38, 2001.

A. Charles, R. Menezes, and R. Tolksdorf. On the implementation of swarmlinda. In *Proceedings of the 42nd Annual Southeast Regional Conference (ACM-SE)*, pages 297--298. ACM Press, 2004.

T. De Wolf and T. Holvoet. Adaptive Behaviour Based on Evolving Thresholds with Feedback. In *Proceedings of the AISB'03 Third Symposium on Adaptive Agents and Multi-Agent Systems (AAMAS)*, pages 91--96, 2003.

G. Di Caro and M. Dorigo. Ant Colonies for Adaptive Routing in Packet-Switched Communications Networks. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature (PPSN V)*, number 1498 in Lecture Notes in Computer Science (LNCS), pages 673--682, London, UK, 1998. Springer-Verlag.

G. Di Marzo Serugendo. On the Use of Formal Specifications as Part of Running Programs. In *Software Engineering for Multi-Agent Systems IV, Research Issues and Practical Applications [the book is a result of SELMAS 2005]*, volume 3914 of *Lecture Notes in Computer Science (LNCS)*, pages 224--237. Springer, 2006.

G. Di Marzo-Serugendo, M.-P. Gleizes, and A. Karageorgos. Self-Organisation and Emergence in MAS: An Overview. *Informatica*, 30(1):45--54, 2005.

M. Dorigo, V. Maniezzo, and A. Colorni. "the ant system: Optimization by a colony of cooperating agents". *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 26(1):29--41, 1996.

B. Farley and W. Clark. "simulation of self-organizing systems by digital computer". *"IEEE Transactions on Information Theory"*, 4(4):76--84, 1954.

S. Forrest. Emergent Computation : Self-Organizing, Collective and Cooperative Phenomena in Natural and Artificial Computing Networks. *Special issue of Physica D*, 1991.

N. Foukia. IDReAM: Intrusion Detection and Response Executed with Agent Mobility Architecture and Implementation. In *Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005)*, pages 264--270. ACM Press, 2005.

N. Foukia and S. Hassas. Managing Computer Networks Security through Self-Organization: A Complex System Perspective. In *Engineering Self-Organising Systems, Nature-Inspired Approaches to Software Engineering*, volume 2977 of *Lecture Notes in Computer Science (LNCS)*, pages 124--138. Springer, 2004.

References (cont.)

M. Gardner. "mathematical games: The fantastic combinations of John Conway's new solitaire game 'Life'". *Scientific American*, 223(4):120--123, 1970.

D. Gelernter. Generative Communication in Linda. *ACM Transactions on Programming Languages and Systems*, 7(1):80--112, 1985.

I. Georgiadis, J. Magee, and J. Kramer. Self-Organising Software Architectures for Distributed Systems. In *Proceedings of the First Workshop on Self-Healing Systems (WOSS'02)*, pages 33--38. ACM Press, 2002.

J.-P. Georgé. *Résolution de problèmes par émergence -- Étude d'un Environnement de Programmation Émergente*. PhD thesis, Université Paul Sabatier - Toulouse III, 2004.

J.-P. Georgé, B. Edmonds, and P. Glize. Making Self-Organising Adaptive Multiagent Systems Work. In F. Bergenti, M.-P. Gleizes, and F. Zombonelli, editors, *Methodologies and Software Engineering for Agent Systems*, pages 319--338. Kluwer Academic Publisher, 2004.

J. Goldstein. Emergence as a Construct : History and Issues. *Journal of Complexity Issues in Organizations and Management*, 1(1), 1999.

P. Grassé. La reconstruction du nid et les interactions inter-individuelles chez les bellicositermes natalenis et cubitermes sp. la théorie de la stigmergie: essai d'interprétation des termites constructeurs. *Insectes Sociaux*, 6:41--83, 1959.

A. Grizard, L. Vercouter, T. Stratulat, and G. Muller. A Peer-to-peer Normative System to Achieve Social Order. In *AAMAS'06 Workshop on Coordination, Organization, Institutions and Norms in agent systems (COIN'06)*, 2006.

D. Hales. Self-Organising, Open and Cooperative P2P Societies - From Tags to Networks. In *Engineering Self-Organising Systems, Methodologies and Applications [revised versions of papers presented at the Engineering Selforganising Applications (ESOA 2004) workshop, held during the Autonomous Agents and Multi-agent Systems conference (AAMAS 2004) in New York in July 2004, and selected invited papers]*, volume 3464 of *Lecture Notes in Computer Science (LNCS)*, pages 123--137. Springer, 2005.

D. Hales. Choose Your Tribe! - Evolution at the Next Level in a Peer-to-Peer Network. In *Engineering Self-Organising Systems, Third International Workshop, ESOA 2005, Utrecht, The Netherlands, July 25, 2005, Revised Selected Papers*, volume 3910 of *Lecture Notes in Computer Science (LNCS)*, pages 61--74. Springer, 2006.

S. Hassas, G. Di Marzo-Serugendo, A. Karageorgos, and C. Castelfranchi. Self-Organising Mechanisms from Social and Business/Economics Approaches. *Informatica*, 30(1):63--71, 2006.

F. Heylighen. *The Encyclopedia of Life-Support Systems*, chapter The Science of Self-Organization and Adaptivity. EOLSS Publishers Co. Lrd, 2001.

M. Jelasity and O Babaoglu. T-Man: Gossip-Based Overlay Topology Management. In *Engineering Self-Organising Systems, Third International Workshop, ESOA 2005, Utrecht, The Netherlands, July 25, 2005, Revised Selected Papers*, volume 3910 of *Lecture Notes in Computer Science (LNCS)*, pages 1--15. Springer, 2006.

H. Karuna, P. Valckenaers, B. Saint Germain, P. Verstraete, C. B. Zamfirescu, and H. Van Brussel. Emergent Forecasting Using a Stigmergy Approach in Manufacturing Coordination and Control. In *Engineering Self-organizing Systems: Methodologies and Applications*, volume 3464 of *Lecture Notes in Computer Science (LNCS)*, pages 210--226. Springer, 2004.

References (cont.)

L. Kutvonen, T. Ruokolainen, and J. Metso. Interoperability middleware for federated business services in web-Pilarcos. *International Journal of Enterprise Information Systems, Special issue on Interoperability of Enterprise Systems and Applications*, 3(1):1--21, 2007.

C. Langton. Computation at the Edge of Chaos -- Phase Transition and Emergent Computation. *Physica D*, 4, 1990.

G.H. Lewes. *"Problems of Life and Mind"*. Kegan Paul, trench, Turbner, & Co, 1875.

H. Liu, M. Parashar, and S. Hariri. A Component-based Programming Model for Autonomic Applications. In *Proceedings. International Conference on Autonomic Computing (ICAC'04)*, pages 10--17, 2004.

M. Mamei and F. Zambonelli. Programming Stigmergic Coordination with the TOTA Middleware. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005)*, pages 415--422. ACM, 2005.

M. Mamei, F. Zambonelli, and L. Leonardi. Co-Fields: A Physically Inspired Approach to Motion Coordination. *IEEE Pervasive Computing*, 3(2):52--61, 2004.

J.-P. Mano. Self-Organization in Natural System. Invited talk, Technical Forum Group on Self-Organization in MAS, 2004.

J.-P. Mano, C. Bourjot, G. Leopardo, and P. Glize. Bio-inspired Mechanisms for Artificial Self-Organised Systems. *Informatica*, 30(1):55--62, 2006.

P. Marrow, C. Hoile, F. Wang, and E. Bonsma. Evolving Preferences among Emergent Groups of Agents. In *Adaptive Agents and Multi-Agent Systems: Adaptation and Multi-Agent Learning*, volume 2636 of *Lecture Notes in Computer Science (LNCS)*, pages 159--173. Springer, 2003.

H. Maturana and F. Varela. *L'arbre de la connaissance, racines biologique de la compréhension humaine*. Addison-Wesley, 1994.

J-P. Muller. Emergence of collective behaviour: simulation and social engineering. In A. Omicini, P. Petta, and J. Pitt, editors, *Fourth International Workshop on Engineering Societies in the Agents World (ESAW'03)*, volume 3071 of *Lecture Notes in Computer Science (LNCS)*. Springer-Verlag, 2004.

H. V. D. Parunak, S. Brueckner, and J. Sauter. Digital Pheromone Mechanisms for Coordination of Unmanned Vehicles. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'02)*, pages 449--450. ACM Press, 2002.

G. Picard and M.-P. Gleizes. The ADELFE Methodology -- Designing Adaptive Cooperative Multi-Agent Systems. In F. Bergenti, M-P. Gleizes, and F. Zambonelli, editors, *Methodologies and Software Engineering for Agent Systems (Chapter 8)*, pages 157--176. Kluwer Publishing, 2004.

G. Picard and M.-P. Gleizes. Cooperative Self-Organization to Design Robust and Adaptive Collectives. In *Second International Conference on Informatics in Control, Automation and Robotics (ICINCO'05), 14-17 September 2005, Barcelona, Spain, Volume I*, pages 236--241. INSTICC Press, 2005.

G. Picard and P. Glize. Model and Analysis of Local Decision Based on Cooperative Self-Organization for Problem Solving. *Multiagent and Grid Systems -- An International Journal (MAGS)*, 2(3):253--265, 2006.

I. Prigogine and G. Nicolis. *Self Organization in Non-Equilibrium Systems*, chapter III & IV. Wiley and Sons, 1977.

References (cont.)

A. Reitbauer, A. Battino, B. Saint Germain, A. Karageorgos, N. Mehandjiev, and P. Valckenaers. The Mabe Middleware: Extending Multi-Agent Systems to Enable Open Business Collaboration. In *6th IFIP International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services (BASYS)*, volume 159, pages 53--60. Springer, 2004.

C. Reynolds. Flocks, Herds and Schools: A Distributed Behavioral Model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*, pages 25--34. ACM Press, 1987.

J. A. Schumpeter. *The Theory of Economic Development*, volume 9, chapter The Economy as a Whole. Industry and Innovation, 2002.

J. Searle. *The Rediscovery of the Mind*. MIT Press, 1992.

M. Stewart. *The Coevolving Organization*. Decomplexity Associates LtD, 2001.

V. Thomas, C. Bourjot, V. Chevrier, and D. Desor. Hamelin: A Model for Collective Adaptation based on Internal Stimuli. In *From Animal to Animats 8 - Eighth International Conference on the Simulation of Adaptive Behaviour 2004 - SAB'04, Los Angeles, USA*, pages 425--434, 2004.

G. Van de Vijver. Emergence et explication. *Intellectica*, 2(25):185--194, 1997.

V. A. Vittikh and P. O. Skobelev. Multi-Agent Systems for Modelling of Self-Organization and Cooperation Processes. In *13th International Conference on the Application of Artificial Intelligence in Engineering*, pages 91--96, 2002.

J. von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966.

J.-B. Welcomme, M.-P. Gleizes, and R. Redon. A Self-Organising Multi-Agent System Managing Complex System Design Application to Conceptual Aircraft Design. In *International Conference on Complex Open Distributed Systems (CODS), Chengdu, 22/07/2007-24/07/2007*, 2007.

D. Weyns, K. Schelfthout, T. Holvoet, and O. Glorieux. Role Based Model for Adaptive Agents. In *Fourth Symposium on Adaptive Agents and Multi-agent Systems, AISB '04 Convention*, 2004.

S.R. White, J.E. Hanson, I. Whalley, D.M. Chess, and J.O. Kephart. An Architectural Approach to Autonomic Computing. In *Proceedings. International Conference on Autonomic Computing (ICAC'04)*, pages 2--9, 2004.