Self-Organisation & MAS
An Introduction

Multiagent Systems LS
Sistemi Multiagente LS

Andrea Omicini & Luca Gardelli
{andrea.omicini, luca.gardelli}@unibo.it

Ingegneria Due
Alma Mater Studiorum—Università di Bologna a Cesena

Academic Year 2011/2012

Outline

Intuitive Idea of Self-Organisation

- Self-organisation generally refers to the internal process leading to an increasing level of organisation
- *Organisation* stands for relations between parts in term of structure and interactions
- *Self* means that the driving force must be internal, specifically, distributed among components

History of Self-Organisation

- The idea of the spontaneous creation of organisation can be traced back to René Descartes
- According to the literature, the first occurrence of the term Self-Organisation is due to a 1947 paper by W. Ross Ashby [Ashby, 1947]
- Ashby defined a system to be self-organising if it changed its own organisation, rather being changed from an external entity

Elements of Self-Organisation

Increasing order — due to the increasing organisation

Autonomy — interaction with external world is allowed as long as the control is not delegated

Adaptive — suitably responds to external changes

Dynamic — it is a process not a final state

Self-Organisation in Sciences

- Initially ignored, the concept of self-organisation is present in almost every science of complexity, including
  - Physics
  - Chemistry
  - Biology and Ecology
  - Economics
  - Artificial Intelligence
  - Computer Science

Outline

History of Emergence

- Emergence is generally referred as the phenomenon involving global behaviours arising from local components interactions
- Although the origin of the term emergence can be traced back to Greeks, the modern meaning is due to the English philosopher G.H. Lewes (1875)
- With respect to chemical reactions, Lewes distinguished between *resultants* and *emergents*
    - Resultants are characterised only by their components, i.e. they are reducible
    - Conversely, emergents cannot be described in terms of their components

Definition of Emergence

- We adopt the definition of emergence provided in [Goldstein, 1999]

  Emergence [..] refers to the arising of novel and coherent structures, patterns, and properties during the process of self-organisation in complex systems. Emergent phenomena are conceptualised as occurring on the macro level, in contrast to the micro-level components and processes out of which they arise.

Emergence vs. Holism

- Emergence is often, and imprecisely, explained resorting to *holism*
- Holism is a theory summarisable by the sentence *the whole is more than the sum of the parts*
- While it is true that an emergent pattern cannot be reduced to the behaviour of the individual components, emergence is a more comprehensive concept

Properties of Emergent Phenomena

Novelty — unpredictability from low-level components

Coherence — a sense of identity maintained over time

Macro-level — emergence happens at an higher-level w.r.t. to components

Dynamism — arise over time, not pre-given

Ostensive — recognised by its manifestation

Requirements for Emergency

Emergence can be exhibited by systems meeting the following requirements

Non-linearity — interactions should be non-linear and are typically represented as feedback-loops

Self-organisation — the ability to self-regulate and adapt the behaviour

Beyond equilibrium — non interested in a final state but on system dynamics

- Attractors — dynamically stable working state

Outline

Definition of Self-Organisation

Widespread definition of Self-Organisation by [Camazine et al., 2001]

Self-organisation is 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. Moreover, the rules specifying interactions among the system's components are executed using only local information, without reference to the global pattern.

- It is evident that the authors conceive self-organisation as the source of emergence
- This tendency of combining emergence and self-organisation is quite common in biological sciences
- In the literature there is plenty of misleading definitions of self-organisation and emergence [De Wolf and Holvoet, 2005]

Outline

Self-Organisation of Matter

- Self-organisation of matter happens in several fashion
- In magnetisation, spins spontaneously align themselves in order to repel each other, producing and overall strong field
- Bérnard Rolls is a phenomena of convection where molecules arrange themselves in regular patterns because of the temperature gradient

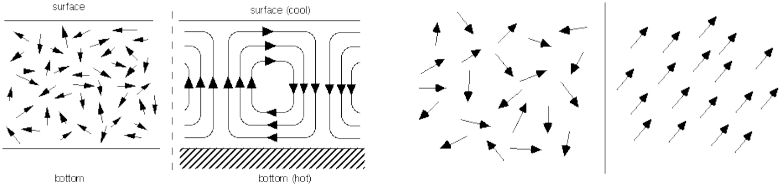


Figure: The left hand side picture display Bérnard Rolls. The right hand side picture display the magnetisation phenomena.

Belousov-Zhabotinsky Reaction I

- Discovered by Belousov in the 1950s and later refined by Zhabontinsky, BZ reactions are a typical example of far from equilibrium system
- Mixing chemical reactants in proper quantities, the solution color or patterns tend to oscillate
- These solutions are referred as *chemical oscillators*
- There have been discovered several reactions behaving as oscillators

Belousov-Zhabotinsky Reaction II

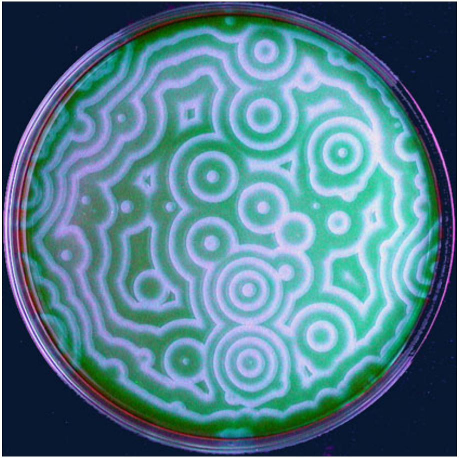


Figure: A snapshot of the Belousov-Zhabotinsky reaction.

Outline

Prey-Predator Systems

- The evolution of a prey-predator systems leads to interesting dynamics
- These dynamics have been encoded in the Lotka-Volterra equation [Solé and Bascompte, 2006]
- Depending on the parameters values the system may evolve either to overpopulation, extinction or periodical evolution

$$\frac{dx}{dt} = x(\alpha - \beta y)$$

$$\frac{dy}{dt} = -y(\gamma - \delta x)$$

Figure: The Lotka-Volterra equation.

Lotka-Volterra Equation

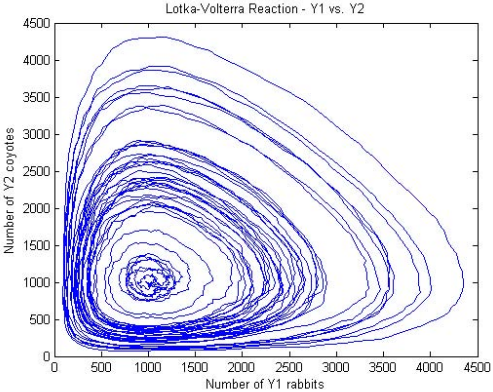


Figure: A chart depicting the state space defined by the Lotka-Volterra equation.

Outline

Synchronised Flashing in Fireflies I

- Some species of fireflies have been reported of being able to synchronise their flashing [Camazine et al., 2001]
- Synchronous flashing is produced by male during mating
- This synchronisation behaviour is reproducible using simple rules
    - Start counting cyclically
    - When perceive a flash, flash and restart counting

Synchronised Flashing in Fireflies II

Figure: A photo of fireflies flashing synchronously.

Schools of Fishes

Figure: School of fishes exhibit coordinated swimming: this behaviour can be simulated based on speed, orientation and distance perceptions [Camazine et al., 2001].

Flocks of Birds

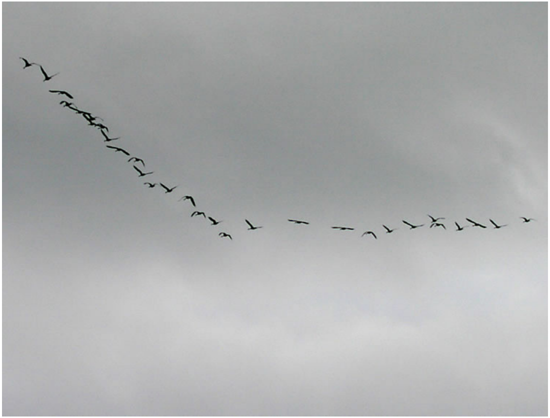


Figure: The picture displays a flock of geese: this behaviour can be simulated based on speed, orientation and distance perceptions [Camazine et al., 2001].

Outline

Insects Colonies

- Behaviours displayed by social insects have always puzzled entomologist
- Behaviours such as nest building, sorting, routing were considered requiring elaborated skills
- For instance, termites and ants build very complex nests, whose building criteria are far than trivial, such as inner temperature, humidity and oxygen concentration

Termites Nest in South Africa

Figure: The picture displays the Macrotermes michealseni termite mound of southern Africa.

Definition of Stigmergy

- In a famous 1959 paper [Grassé, 1959], Grassé proposed an explanation for the coordination observed in termites societies

  The coordination of tasks and the regulation of constructions are not directly dependent from the workers, but from constructions themselves. The worker does not direct its own work, he is driven by it. We name this particular stimulation stigmergy.

Elements of Stigmergy

- Nowadays, stigmergy refers to a set of coordination mechanisms mediated by the environment
- For instance in ant colonies, chemical substances, namely *pheromone*, act as markers for specific activities
- E.g. the ant trails between food source and nest reflect the spatial concentration of pheromone in the environment

Trail Formation in Ant Colonies

Figure: The picture food foraging ants. When carrying food, ants lay pheromone, adaptively establishing a path between food source and the nest. When sensing pheromone, ants follow the trail to reach the food source.

Simulating Food Foraging

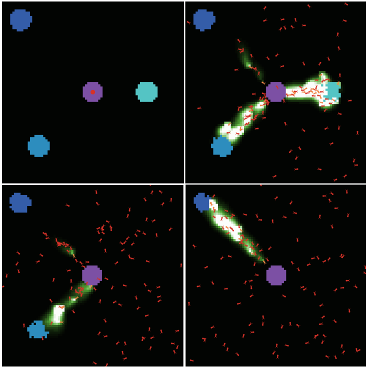


Figure: The snapshots display a simulation of food foraging ants featuring a nest and three food sources. Ants find the shortest path to each sources ad consume first the closer sources. When no longer reinforced, the pheromone eventually evaporates.

Stigmergy and the Environment

- In stigmergy, the environment play a fundamental roles, collecting and evaporating pheromone
- In its famous book [Resnick, 1997], Resnick stressed the role of the environment

The hills are alive. The environment is an active process that impacts the behavior of the system, not just a passive communication channel between agents.

Outline

Swarm Intelligence

- Is a problem solving approach inspired by collective behaviours displayed by social insects [Bonabeau et al., 1999, Bonabeau and Théraulaz, 2000]
- It is not a uniform theory, rather a collection of mechanisms found in natural systems having applications to artificial systems
- Applications of Swarm Intelligence include a variety of problems such as task allocation, routing, synchronisation, sorting
- In Swarm Intelligence, the most successful initiative is Ant Colony Optimisation

ACO: Ant Colony Optimisation

- ACO [Dorigo and Stützle, 2004] is a population-based metaheuristic that can be used to find approximate solutions to difficult optimisation problems
- A set of software agents called artificial ants search for good solutions to a given optimisation problem
- To apply ACO, the optimisation problem is transformed into the problem of finding the best path on a weighted graph
- ACO provided solutions to problems such as VRP-Vehicle Routing Problem, TSP-Travelling Salesman Problem and packet routing in telecommunication networks

Amorphous Computing

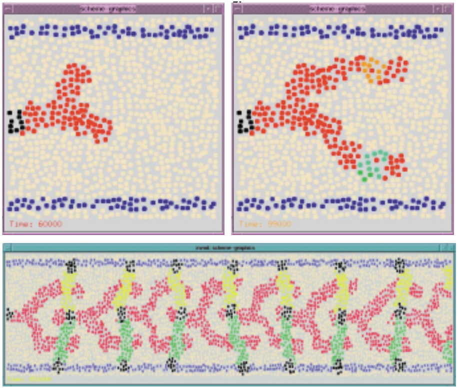


Figure: An amorphous computing [Abelson et al., 2000] medium is a system of irregularly placed, asynchronous, locally interacting identical computing elements.

Autonomic Computing

- An industry driven research field initiated by IBM [Kephart and Chess, 2003], mostly motivated by increasing costs in systems maintenance
- Basic idea: applying self-organising mechanisms found in human nervous system to develop more robust and adaptive systems
- Applications range from a variety of problems such as power saving, security, load balancing

Outline

Robocup I

By the year 2050, develop a team of fully autonomous humanoid robots that can win against the human world soccer champion team.

- Robocup objective consists in pushing robotics research applying the techniques developed to eventually win soccer match
- Robocup matches are organised in leagues reflecting different robot capabilities
- Self-organising techniques are extensively applied since the robots have to be autonomous rather than remotely controlled

Robocup II

Figure: A few robots that have participated to Robocup 2006 edition.

SWARM-BOTS

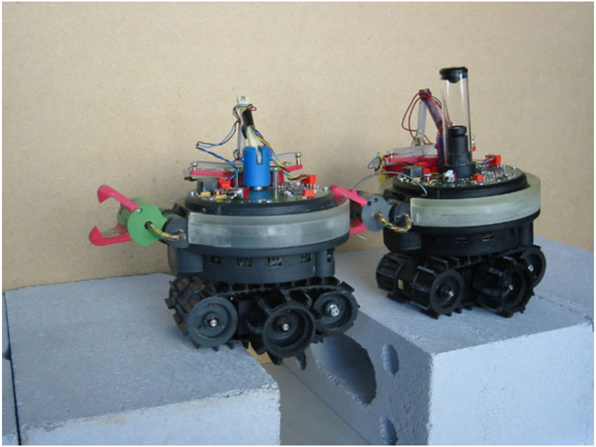


Figure: SWARM-BOTS [Dorigo et al., 2005] was a project funded by European Community tailored to the study of self-organisation and self-assembly of modular robots.

AGV – Automated Guided Vehicles

- Stigmergy has been successfully applied to several deployments of Automated Guided Vehicles [Weyns et al., 2005, Sauter et al., 2005]
- Basically, the AGVs are driven by digital pheromones fields in the same way ants perform food-foraging

Figure: Various pictures of AGVs

Outline

MAS 4 SOS

- Is the agent paradigm the right choice for modelling and developing SOS?
- In order to answer this question we have to compare requirements for SOS with features of MAS

SOS Requirements

- From previous discussion on self-organisation and emergence we can identify this basic requirements list
  - Autonomy and encapsulation of behaviour
  - Local actions and perceptions
  - Distributed environment supporting interactions
  - Support for organisation and cooperation concepts

MAS Checklist

- It is easy to recognise that the agent paradigm provides suitable abstractions for each aspect
- Indeed, MAS are currently the reference for both self-organisation modelling and engineering
- In self-organisation literature not having a background in computer science, it is often the case that the term agent is used with a different meaning
- For instance, in biology and chemistry complex chemical compounds are often called agents without actually referring to the agent paradigm

Outline

Current MAS Methodologies

- Most MAS methodologies were developed because of the need to address specific issues
- For instance Gaia was initially concerned more with intra-agent aspect, while SODA dealt with aspects at the society level
- Engineering methodologies are related to the paradigm in use
- Being interested in SOSs, we need a methodology that supports the basic requirements previously identified

MAS Methodologies for SOS

- Unfortunately there are only a few methodologies soundly supporting organisation and environmental aspects [Molesini et al., 2007]
- The ADELFE methodology is a proposal for Adaptive MAS where properties emerges by self-organisation [Bernon et al., 2004]
- Although considering cooperation and environmental issues of self-organisation, in our opinion ADELFE provide no pragmatic approach for the engineering of emergence

Designing Self-Organising Emergent Systems

- In developing artificial self-organising systems displaying emergent properties we identify two main issues [Gardelli et al., 2007a, Gardelli et al., 2007b]
  1. How do we design individual agent behaviour that collectively produce the target emergent property? : Due to non-linearities both in agent behaviour and environmental dynamics devising a strategy that eventually leads to the target property is a very difficult problem.
  2. How do we evaluate a specific solution and provide actual guarantees of its quality? : Because of dependability requirement, we cannot deploy a system without having profiled the possible evolutions and framed the working environmental conditions.

Outline

Intro

- In the rest of the seminar we describe our approach for the engineering self-organising MAS with emergent properties
- In particular we consider issues related both to workflow and tools
- The material presented from now on is mostly based on [Gardelli et al., 2007a, Gardelli et al., 2007b]
- We now start considering the two previous issues, one at a time

Issue 1: Forward vs. Reverse Engineering

- How do we design individual agent behaviour that collectively produce the target emergent property?
- It is generally acknowledged that forward engineering of emergent properties is feasible only for small/trivial problems
- Indeed, most of the artificial self-organising systems have been inspired by natural systems

Issue 1: Inspiration

- Although pervasive, "inspiration" process is not a scientific approach and it is hardly reproducible
- We need a way to map computer science problems into successful natural strategies
- Only recently, it has been recognised the need of a more formal approach when designing SO MAS: a few proposal involve design patterns [Babaoglu et al., 2006, De Wolf and Holvoet, 2007, Gardelli et al., 2007c]

Issue 1: Design Patterns

- Initially introduced in architectural engineering, design patterns have been popularised in computer science in the 1990s along with the object-oriented paradigm [Gamma et al., 1995]
- A design pattern provide a reusable solution to a recurrent problem in a specific domain
- In our context design patterns are a viable approach to encode successful solution provided by natural systems to computer science problems [Babaoglu et al., 2006, De Wolf and Holvoet, 2007, Gardelli et al., 2007c]
- Although there have been already proposed several patterns, we are confident that we will not find a suitable pattern for every computer science problem: we will discuss it later when dealing with Issue 2

Issue 1: Feedback Loop

- Self-organisation and Emergence involve the existence of a feedback loop
- Such feedback loop is often produced by a functional coupling between agents and the environment
- E.g. consider the ants depositing pheromone while the environment evaporates it

Issue 1: Architectural Pattern I

- When designing a SO MAS according to the Agents & Artifacts metamodel [Ricci et al., 2006] we identify a recurrent architectural solution
- Since, it is often the case that the agent environment is partially or completely given, such as in case of legacy resources, we do not have complete control over the environment
- Hence, being difficult to embed self-organisation into artifacts, we introduce environmental agents whose role is to close the feedback loop between agents properly managing artifacts behaviour
- Furthermore, environmental agents allow a finer control isolating normal behaviour from the one responsible of emergent properties

Issue 1: Architectural Pattern II

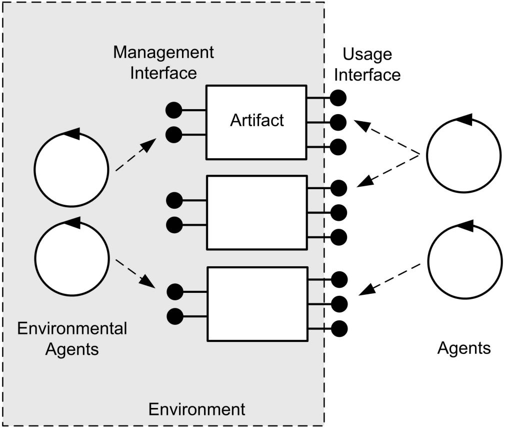


Figure: The architectural pattern featuring environmental agents encapsulating self-organising behaviour and managing artifacts.

Issue 1: Summarising

- Forward engineering of emergent properties is not feasible, hence we rely on the existence of a natural system providing a suitable solution
- Such solution should be encoded as a design pattern eventually leading to the creation of a coherent pattern catalogue
- In particular the design pattern should provide behaviours for the three roles identifies in the architectural pattern: agents, artifacts and environmental agents

Issue 2: Towards a Workflow

- How do we evaluate a specific solution and provide actual guarantees of its quality?
- In order to fulfill this issue we promote the following iterative engineering process
  1. Modelling
  2. Simulation
  3. Verification
  4. Tuning (if needed then back to step 2)

Issue 2: Exploiting Formal Tools

- Since we are going to perform several tasks on a given model we promote the use of formal tools
- Formal languages allow the specification of selective and unambiguous models and provide a solid basis for automatic processing
- Hence, having a model expressed in a suitable formal language we can

    1. run simulations by specifying only operating parameters
    2. verify the system by model-checking just providing the properties in a suitable temporal logic

Workflow: Modelling

- During the modelling phase we have, according to the architectural pattern, identify the roles of each entity, namely agents, artifacts and environmental agents
- The individual behaviour is to be found within the design pattern catalogue
- Modifications to the pattern may be required to fit the actual requirements: this is a non-trivial step and requires expertise in the domain
- In this phase the model should not be too detailed, rather reflect the abstract architecture of the system: indeed a fine-grained model can prevent further automatic processing

Workflow: Simulation

- Simulation allows us to qualitatively preview the global system dynamics
- Before running the simulation we have to provide working parameters for agents and artifacts, while parameters set for environmental agents is our unknown variable
- Needless to say that in order for the simulation results to be valid parameters should reflect the actual deployment conditions
- Although the use of simulation is a common practice in system engineering, it is almost unused in software development
- In self-organisation literature, the need for simulation has been recognised only recently [Gardelli et al., 2006] [Gardelli et al., 2007a] [Bernon et al., 2006] [De Wolf et al., 2006]

Workflow: Verification

- Simulation alone does not provide sound guarantees because of incompleteness
- Conversely, model checking [Edmund M. Clarke et al., 1999] is a formal technique for verifying automatically the properties of a target system against its model
- The model to be verified is expressed in a formal language, typically in a transition system fashion
- Then, properties to be verified are formalised using a variant of temporal logic depending on the current model
- The main drawback of model checking is dependence upon model state space which grows very quickly, becoming unfeasible

Workflow: Tuning

- If the current system model does not meet requirements we have to tune its parameters
- This implies a further cycle, of simulation-verification-tuning
- If the results display discrepancies with requirements we may consider also altering the model

Workflow: Tools

- In order to ease the workflow we need a tool supporting the whole process
- The tool must meet the following requirements
  - provide a formal modelling language allowing to express stochastic aspects
  - provide a built-in stochastic simulator able to run directly from the specified model
  - provide a built-in probabilistic model checker and support the specifications of temporal logic properties

Tools: PRISM

- Among the various available tools we selected PRISM – Probabilistic Symbolic Model Checker developed at University of Birmingham [PRISM, 2007]
- PRISM language allows the specification of models in a transition-system fashion
- The built-in stochastic simulator is very simple but has plotting and exporting capabilities, although more sophisticated tools would have been appreciated
- The built-in probabilistic model checker is very robust: it provides alternative engines and allows the specification of properties both in PCTL – Probabilistic Computational Tree Logic and CSL – Continuous Stochastic Logic

Outline

Problem Statement

- Provided a networked set of nodes not fully connected where each node hosts a certain amount of data items
- Given that each node knows only (i) the number of local items, and (ii) the neighbouring nodes, while has no information about network size and total amount of items
- Devise a self-organising strategy for implementing a plain diffusion strategy that eventually leads the system to a state where each node has the same amount of items

Reference Network Topology

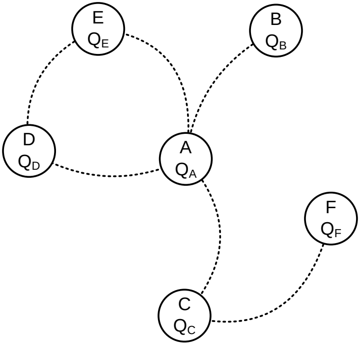


Figure: The reference topology: starting from state $A = 36, B = C = D = E = F = 0$ the system must evolve into $A = B = C = D = E = F = 6$.

Equivalent A&A Topology

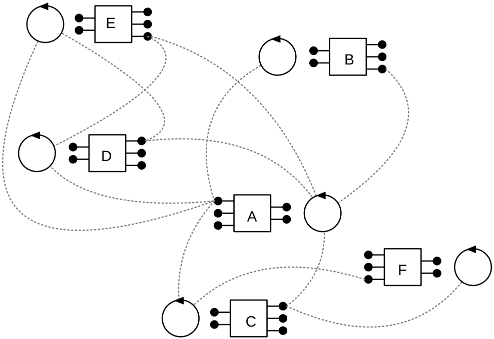


Figure: Notice that this topology is equivalent to the previous one.

Modelling

- We have to provide a strategy for environmental agents that exchanging items with neighbouring artifacts based on local information eventually produce the desired dynamics
- The key is the dynamical equilibrium established by agents exchanging items at different rates: if the exchange rates are identical the situation remains statistically unchanged
- Agents have to exchange items proportionally to the local number of items, i.e. working faster when having large number of items and slower in the other case
- Furthermore, agents should exchange items proportionally to the number of neighbouring nodes: hubs have to work faster to avoid congestion!

PRISM Model

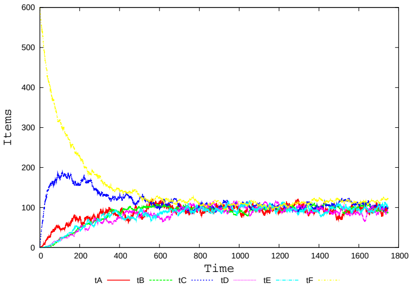
- We describe the model using the PRISM language in order to allow further automatic elaborations
- PRISM language define a transition system

```
module agentA
[] tA > 0 & tB < MAX & tC < MAX & tD < MAX ->
rA : (tA'=tA-1) & (tB'=tB+1) +
rA : (tA'=tA-1) & (tC'=tC+1) +
rA : (tA'=tA-1) & (tD'=tD+1) +
rA : (tA'=tA-1) & (tE'=tE+1);
endmodule
```

Figure: The code snippet show the description of the agent hosted by the hub, node A.

Simulation

- Providing values for system parameters we can run simulations directly from PRISM



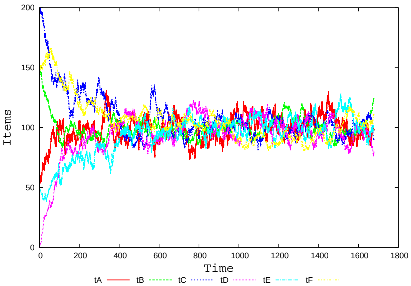


Figure: Two sample simulations from different initial states (left) all items in one node (right) almost sorted

PRISM Model Checking

- Which is the steady-state probability for the node X to contain Y items?: using the PRISM syntax for CSL properties $S =? [tA = Y]$

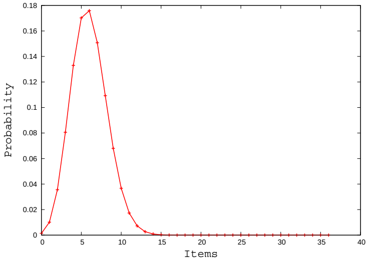


Figure: The chart displays the distribution of the probability for a node to contain a specific number of items: further experiments show that the chart is the same for each node.

Tuning

- Is the probability of reaching the dynamic equilibrium condition within 200 time units greater or equals to 90% ?: using the PRISM syntax for PCTL properties $P >= 0.9\ [true\ U <= 200\ tB = 6]$ for the node tB

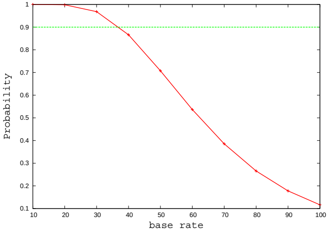


Figure: The chart displays the probability values for the node tB varying base rate parameter: we can guess that the desired value is within the range 30..40.

Bibliography I

Abelson, H., Allen, D., Coore, D., Hanson, C., Homsy, G., Thomas F. Knight, J., Nagpal, R., Rauch, E., Sussman, G. J., and Weiss, R. (2000).
Amorphous computing.
Commununications of the ACM, 43(5):74–82.

Ashby, W. R. (1947).
Principles of self-organizing dynamic systems.
Journal of General Psychology, 37:125–128.

Bibliography II

Babaoglu, O., Canright, G., Deutsch, A., Di Caro, G. A., Ducatelle, F., Gambardella, L. M., Ganguly, N., Jelasity, M., Montemanni, R., Montresor, A., and Urnes, T. (2006).
Design patterns from biology for distributed computing.
Transactions on Autonomous and Adaptive Systems (TAAS), 1(1):26–66.

Bernon, C., Camps, V., Gleizes, M.-P., and Picard, G. (2004).
Designing agents' behaviors and interactions within the framework of ADELFE methodology.
In *Engineering Societies in the Agents World*, volume 3071 of *LNCS (LNAI)*, pages 311–327. Springer.
4th International Workshops, ESAW 2003, London, UK, October 29-31, 2003, Revised Selected and Invited Papers.

Bibliography III

Bernon, C., Gleizes, M.-P., and Picard, G. (2006).
Enhancing self-organising emergent systems design with simulation.
In *Seventh International Workshop on Engineering Societies in the Agents World (ESAW'06)*, Dublin, Ireland. University College Dublin (UCD).

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999).
Swarm Intelligence: From Natural to Artificial Systems.
Santa Fe Institute Studies in the Sciences of Complexity. Oxford University Press, 198 Madison Avenue, New York, New York 10016, United States of America.

Bibliography IV

Bonabeau, E. and Théraulaz, G. (2000).
Swarm smarts (behavior of social insects as model for complex systems).
Scientific American, 282(3):72–79.

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2001).
Self-Organization in Biological Systems.
Princeton Studies in Complexity. Princeton University Press, 41 William Street, Princeton, New Jersey 08540, United States of America.

Bibliography V

De Wolf, T. and Holvoet, T. (2005).
Emergence versus self-organisation: Different concepts but promising when combined.
In Brueckner, S., Di Marzo Serugendo, G., Karageorgos, A., and Nagpal, R., editors, *Engineering Self Organising Systems: Methodologies and Applications*, volume 3464 of *LNCS (LNAI)*, pages 1–15. Springer.

Bibliography VI

De Wolf, T. and Holvoet, T. (2007).
Design patterns for decentralised coordination in self-organising emergent systems.
In Brueckner, S., Hassas, S., Jelasity, M., and Yamins, D., editors, *Engineering Self-Organising Systems*, volume 4335 of *LNCS*, pages 28–49. Springer.
Fourth International Workshop, ESOA 2006, Future University-Hakodate, Japan, 2006, Revised Selected Papers.

Bibliography VII

De Wolf, T., Holvoet, T., and Samaey, G. (2006).
Development of self-organising emergent applications with simulation-based numerical analysis.
In Brueckne, S. A., Di Marzo Serugendo, G., Hales, D., and Zambonelli, F., editors, *Engineering Self-Organising Systems*, volume 3910 of *LNCS*, pages 138–152. Springer.
Third International Workshop, ESOA 2005, Utrecht, The Netherlands, July 2005, Revised Selected Papers.

Dorigo, M. and Stützle, T. (2004).
Ant Colony Optimization.
MIT Press, Cambridge, MA.

Bibliography VIII

Dorigo, M., Tuci, E., Mondada, F., Nolfi, S., Deneubourg, J.-L., Floreano, D., and Gambardella, L. M. (2005).
The SWARM-BOTS project.
Künstliche Intelligenz, 4/05:32–35.
Also available at http://www.swarm-bots.org as IRIDIA Technical Report No. TR/IRIDIA/2005-018.

Edmund M. Clarke, J., Grumberg, O., and Peled, D. A. (1999).
Model Checking.
The MIT Press.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995).
Design patterns : elements of reusable object-oriented software.
Professional Computing. Addison-Wesley, One Lake Street, Upper Saddle River, NJ, 07458, USA.

Bibliography IX

Gardelli, L., Viroli, M., Casadei, M., and Omicini, A. (2007a).
Designing self-organising environments with agents and artifacts: A simulation-driven approach.
International Journal of Agent-Oriented Software Engineering, 2(2). In Press.

Gardelli, L., Viroli, M., Casadei, M., and Omicini, A. (2007b).
Designing self-organising MAS environments: The collective sort case.
In Weyns, D., Parunak, H. V. D., and Michel, F., editors, *Environments for Multi-Agent Systems III*, volume 4389 of *LNAI*, pages 254–271. Springer.
3rd International Workshop (E4MAS 2006), Hakodate, Japan, 8 May 2006. Selected Revised and Invited Papers.

Bibliography X

Gardelli, L., Viroli, M., and Omicini, A. (2006).
On the role of simulations in engineering self-organising mas: The case of an intrusion detection system in tucson.
In Brueckner, S. A., Di Marzo Serugendo, G., Hales, D., and Zambonelli, F., editors, *Engineering Self-Organising Systems*, volume 3910 of *LNAI*, pages 153 – 166. Springer Berlin / Heidelberg.
Third International Workshop, ESOA 2005, Utrecht, The Netherlands, July 25, 2005, Revised Selected Papers.

Bibliography XI

Gardelli, L., Viroli, M., and Omicini, A. (2007c).
Design patterns for self-organising systems.
In Burkhard, H.-D., Lindemann, G., Verbrugge, R., and Varga, L. Z., editors, *Multi-Agent Systems and Applications V*, volume 4696 of *LNCS (LNAI)*, pages 123–132. Springer, Heidelberg.
5th International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2007, Leipzig, Germany, September 25–27, 2007.

Goldstein, J. (1999).
Emergence as a construct: History and issues.
Emergence, 1(1):49–72.

Bibliography XII

Grassé, P.-P. (1959).
La reconstruction du nid et les coordinations interindividuelles chez bellicositermes natalensis et cubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs.
Insectes Sociaux, 6(1):41–80.

Kephart, J. O. and Chess, D. M. (2003).
The vision of autonomic computing.
Computer, 36(1):41–50.

Molesini, A., Omicini, A., and Viroli, M. (2007).
Environment in agent-oriented software engineering methodologies.
International Journal on Multiagent and Grid Systems.
In Press. Special Issue on Engineering Environments for Multiagent Systems.

Bibliography XIII

PRISM (2007).
PRISM: Probabilistic symbolic model checker.
Developed at University of Birmingham. Version 3.1.1 available online at http://www.prismmodelchecker.org/.

Resnick, M. (1997).
Turtles, termites, and traffic jams: explorations in massively parallel microworlds.
MIT Press, Cambridge, Massachusetts 02142, USA.

Bibliography XIV

Ricci, A., Viroli, M., and Omicini, A. (2006).
Programming MAS with artifacts.
In Bordini, R. P., Dastani, M., Dix, J., and El Fallah Seghrouchni, A., editors, *Programming Multi-Agent Systems*, volume 3862 of *LNAI*, pages 206–221. Springer.
3rd International Workshop (PROMAS 2005), AAMAS 2005, Utrecht, The Netherlands, July 26, 2005. Revised and Invited Papers.

Bibliography XV

Sauter, J. A., Matthews, R. S., Parunak, H. V. D., and Brueckner, S. (2005).
Proceedings of the 4th international joint conference on autonomous agents and multiagent systems (aamas 2005).
In Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M. P., and Wooldridge, M., editors, *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005)*, pages 903–910, Utrecht, The Netherlands. ACM Press.

Solé, R. V. and Bascompte, J. (2006).
Self-Organization in Complex Ecosystems.
Number 42 in Monographs in population Biology. Princeton University Press, 41 William Street, Princeton, New Jersey 08540, United States of America.

Bibliography XVI

Weyns, D., Schelfthout, K., Holvoet, T., and Lefever, T. (2005).
Proceedings of the 4th international joint conference on autonomous agents and multiagent systems (aamas 2005).
In Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M. P., and Wooldridge, M., editors, *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005)*, pages 67–74, Utrecht, The Netherlands. ACM Press.

Self-Organisation & MAS
An Introduction

Multiagent Systems LS
Sistemi Multiagente LS

Andrea Omicini & Luca Gardelli
{andrea.omicini, luca.gardelli}@unibo.it

Ingegneria Due
Alma Mater Studiorum—Università di Bologna a Cesena

Academic Year 2011/2012