Alma Mater Studiorum-Università di Bologna
DEIS - Dipartimento di Elettronica, Informatica e Sistemistica
Dottorato di Ricerca in Ingegneria Elettronica, Informatica
e delle Telecomunicazioni – ING/INF-05 – Ciclo XX

Ingegneria di Sistemi Auto-organizzanti con il Paradigma Multiagente

Engineering Self-Organising Systems with the Multiagent Paradigm

Ing. Luca Gardelli
luca.gardelli@unibo.it

Coordinatore: Prof. Ing. Paolo Bassi
Relatore: Prof. Ing. Antonio Natali
Correlatori: Prof. Ing. Andrea Omicini, Dott. Ing. Mirko Viroli

Abstract

The increasing complexity inherent to modern computational systems is a challenge for the software designer. In particular, aspects such as pervasiveness, heterogeneity, and distributedness contrast with the increasing requirements on dependability. The theory of self-organisation deals with natural systems made of several entities that, through local interactions, are able to re-organise upon environmental changes. After a brief discussion of basic concepts and principles, we describe our approach to the engineering of self-organising multiagent systems. In particular, we focus on methodological aspects to analyse and profile self-organisation mechanisms as well as emergent properties. To this purpose, we rely on formal methods at all the stages of the approach, namely, modelling, simulation, verification, and tuning. To clarify the approach we discuss our solution to the Plain Diffusion problem.

Outline

- Motivation
- Background
  - Self-Organisation and Emergence
  - Multiagent Paradigm and A&A Metamodel
- Our Contribution
  1. Design Patterns for Self-Organising Systems
  2. Methodological Approach for Engineering Self-Organising Multiagent Systems
  3. Engineering a Simple Distributed Strategy: the Case of Plain Diffusion
- Concluding Remarks

Motivation

Why Studying SOSs?

- The increasing complexity and requirements of dependability of modern computational systems calls for more robust techniques
- Despite their very complexity, self-organising systems solve problems in a very elegant and robust way, often producing the solution by emergence

Why Studying SOSs?

- Mechanisms observed in Nature have been successfully applied to distributed computing systems, e.g. routing and coordination of autonomous guided vehicles
- IBM and NASA are investing on self-organisation research for their projects, respectively, Autonomic Computing and the SWARM mission

Why MAS for SOS?

- The Multiagent paradigm seems the most appropriate for modelling and designing self-organising systems because of the natural mapping between the abstractions
- SOS components maps to agents, while the environment dynamics are realised using environmental abstractions, i.e. artefacts in the A&A metamodel

Main Challenges of SOSs

- There is no methodology to devise the set of rules that let emerge only the desired properties
- Then, how can we provide guarantees about the emergence of specific properties?
- Our research activities evolved around these two issues

Background

Self-Organisation

- The term self-organisation suggests the idea of internal processes creating and supporting organisation
- The first use of the term with its modern meaning is due to a 1947 paper by the English psychiatrist William Ross Ashby

The ability of a system to change its own internal organisation, rather being changed from an external force.

Key Features of Self-Organisation

- Autonomy: organisation changes are produced by the interactions of the internal components
- Organisation: it is not restricted to structural organisation, e.g. it may also be functional
- Dynamic: it is a process, not a final state
- Adaptive: it is able to react to changes in its environment, eventually re-organising itself

Emergence

- The notion of emergence is linked to novelty and an abstraction gap between the system components and the observed property
- The modern meaning can be traced back to the philosophers of the British Emergentism, XIX century
- Early formulations were linked to the observation of properties in chemical compounds that were irreducible to their components

Emergence

- A recent definition of Emergent System by J.-P. Muller:
  1. a system of entities in interaction whose expression of the states and dynamics is made in an ontology or theory D,
  2. the production of a phenomenon, which could be a process, a stable state, or an invariant, which is necessarily global regarding the system of entities,
  3. the interpretation of this global phenomenon either by an observer or by the entities themselves via an inscription mechanism in another ontology or theory D'.

Example: Termites Mound

Example: Ants

Multiagent Paradigm

- Systems are conceived in terms of agents and environment
  – Agents are autonomous entities driven by their internal goals: agents are situated within an environment, and can perceive and affect the environment
  – The environment, beyond supporting agents lifecycle, provides services to agents: services are encoded in terms of environmental abstraction

The A&A Metamodel

- There exist different interpretations of the multiagent paradigm
- In particular, we adopt the Agent&Artefact metamodel were artefacts concretise the notion of environmental abstraction
- An artefact provides services to agents that are accessible through a usage interface

Artefacts

- Artefacts encapsulate those activities that do not require to be characterised as goal oriented
- Artefacts embody the portion of the environment supporting MAS activities
- Artefacts can be used to encapsulate legacy resources providing a unified interface

Our Contribution:
1) Design Patterns

Inspiration…

- Currently, there is no sistematic way to design entities behaviour to produce a target global property by emergence
- Hence, designers typically gather inspiration from known models of natural systems
- Inspiration is not a reliable scientific practice!
- Flipping the pages of Biology manuals is a time consuming and error-prone task!

Design Patterns?

- We propose the use of design patterns to encode natural strategies that have been successfully applied to computational systems
- The use of design patterns has been popularised in computer science with the object-oriented paradigm
- It allows to reuse successful strategies, reducing errors and providing a shared ontology for designers

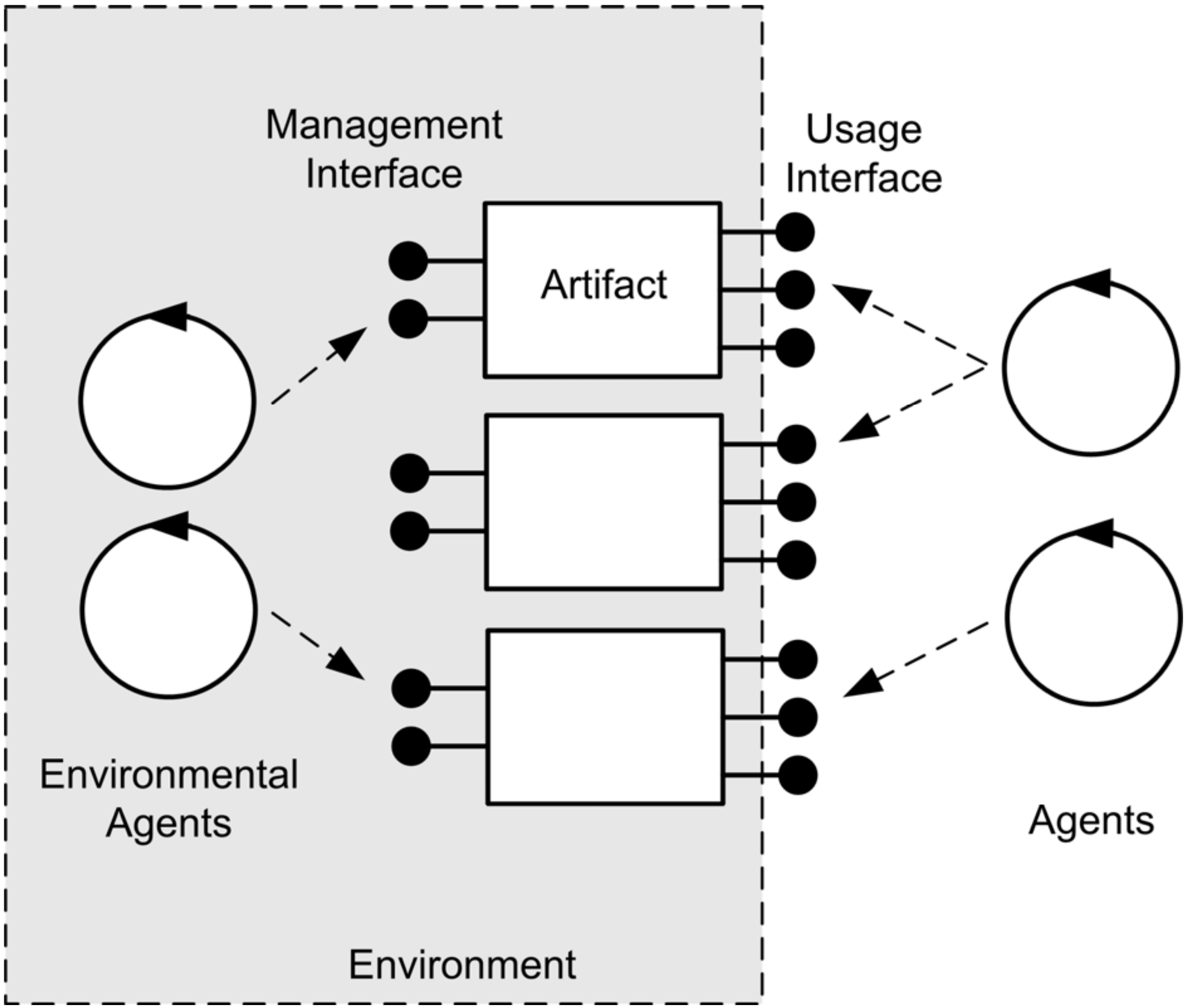
Our Reference Architectural Pattern

- Self-organisation and emergence processes involve both the agents and the environment
- Indeed, the feedback loop responsible both for self-organisation and emergence crosses the agent boundaries
- Self-organisation mechanisms are partially encoded in the agents and in the artefacts

Our Reference Architectural Pattern

- Unfortunately, when working with legacy systems, (that are effectively wrapped by artefacts) we may have little or no control over the environment
- Hence, we introduce the notion of environmental agent which is responsible for managing the artefact in such a way to achieve the desired self-organising behaviour
- This pattern is similar to the one used in Autonomic Computing

Our Reference Architectural Pattern



Our Contribution:
2) Methodology

A Methodological Approach

- We propose a methodological approach for the engineering of self-organising MAS structured according to the previous architectural pattern
- The approach is not a complete methodology, rather a collection of best practices to lead the early design stage
- The approach is articulated in 4 steps
    1. Modelling
    2. Simulation
    3. Verification
    4. Tuning

Modelling

- Propose a model based on existing patterns
- It is likely that we cannot identify the exact desired behaviour among the existing ones
- Although, experiences show that it is possible to adapt existing ones to the designer needs
- Non deterministic aspects are encoded using a stochastic characterisation
- The model specifies the roles of the entities, i.e. agents, environmental agents, artefacts

Simulation

- The stochastic models developed in the previous stage are then evaluated by simulation
- Simulation allows us to analyse system dynamics before committing to the actual design
- We mainly observe if the system displays the target emergent properties

Verification

- In this stage we perform more in-depth and careful analysis of the model
- The verification process allows us to characterise more precisely the observed phenomena
- To this purpose, we mostly rely on formal techniques, specifically, stochastic model checking

Tuning

- In this stage we tweak model parameters in order to reach the desired global dynamics
- Tuning may end up to a set of suitable parameters or providing evidence of unfeasibility
- In case of unfeasibility then it is necessary resort to a different model

Formal Tools

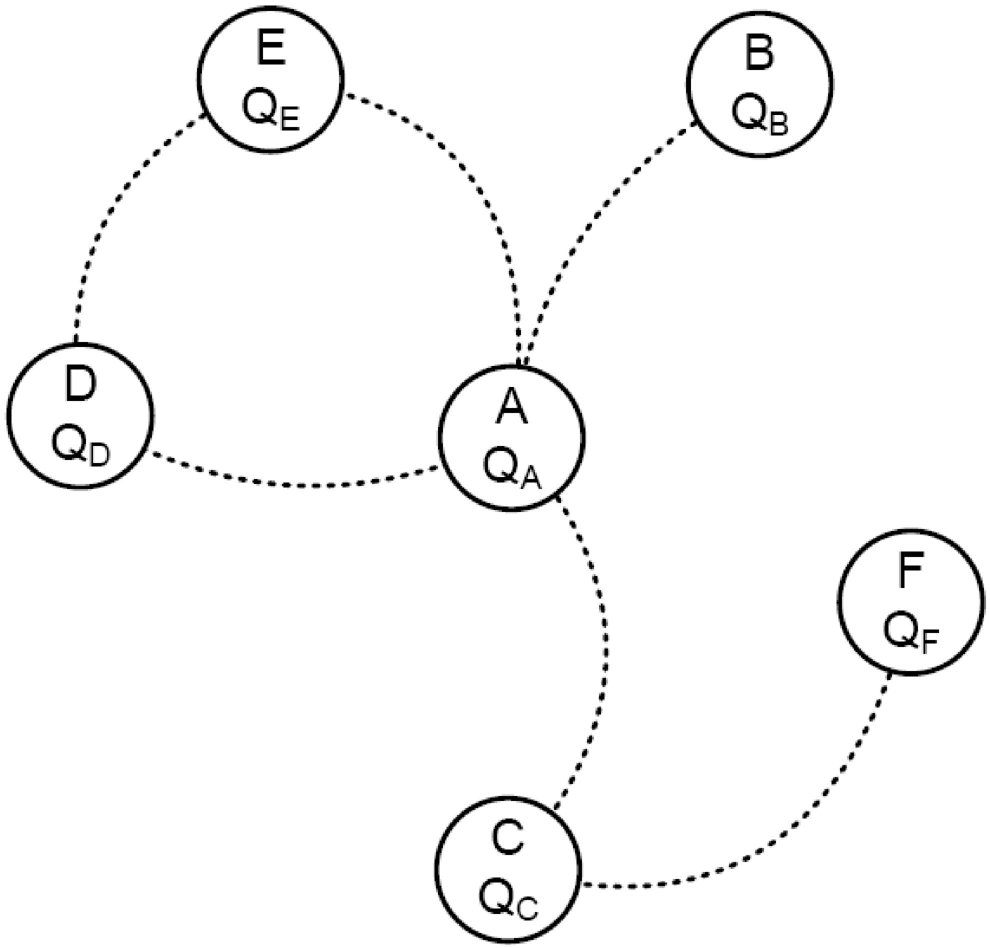
- We use formal tools throughout the whole process
- They provide unambiguous specifications and allow the automation of several analysis tasks without the need of recoding
- We evaluated several formalisms and tools, but we currently rely on the use of PRISM the Probabilistic Symbolic Model Checker developed at University of Birmingham

Our Contribution:
3) Case Study

Problem Statement

- Consider a networked set of nodes having an arbitrary topology and where each node is labelled with a non-negative quantity
- We want to devise a strategy that from an arbitrary initial state eventually evolves into a dynamical state where each node is labelled with the same quantity
- The strategy solving this problem is known as Plain Diffusion

Reference Network



Constraints

- Each node must act autonomously
- A node does known
  - The local label
  - The neighbouring nodes
- A node does not known
  - The total number of nodes
  - The labels of the other nodes

Proposed model

- Each node sends an item to each neighbour with a rate proportional to the local number of items
- Hence, for each neighbouring **artefact**, the **environmental agent** moves an item from local to remote artefact with a rate proportional to the local artefact content

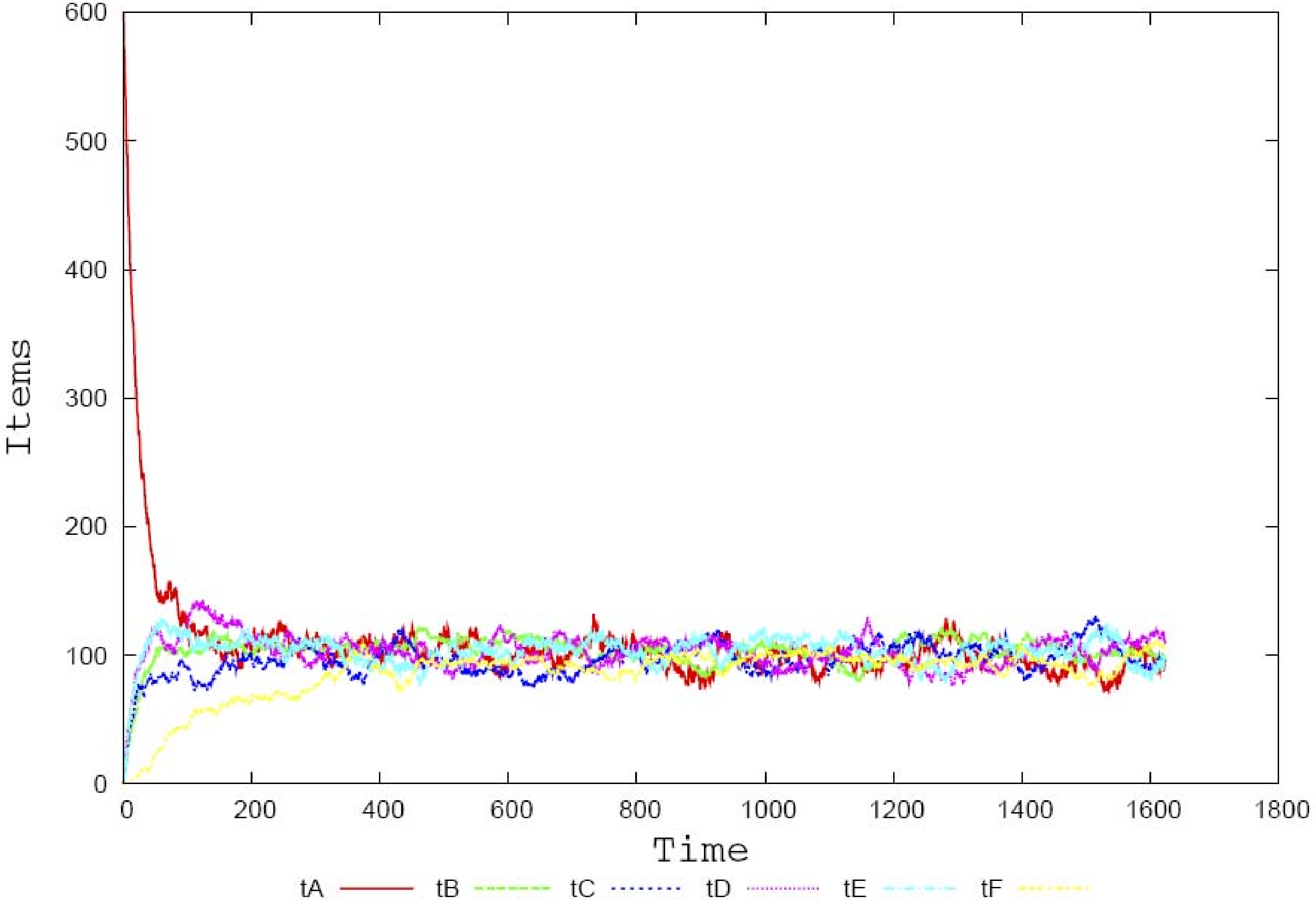
Example of PRISM encoding

```
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
```

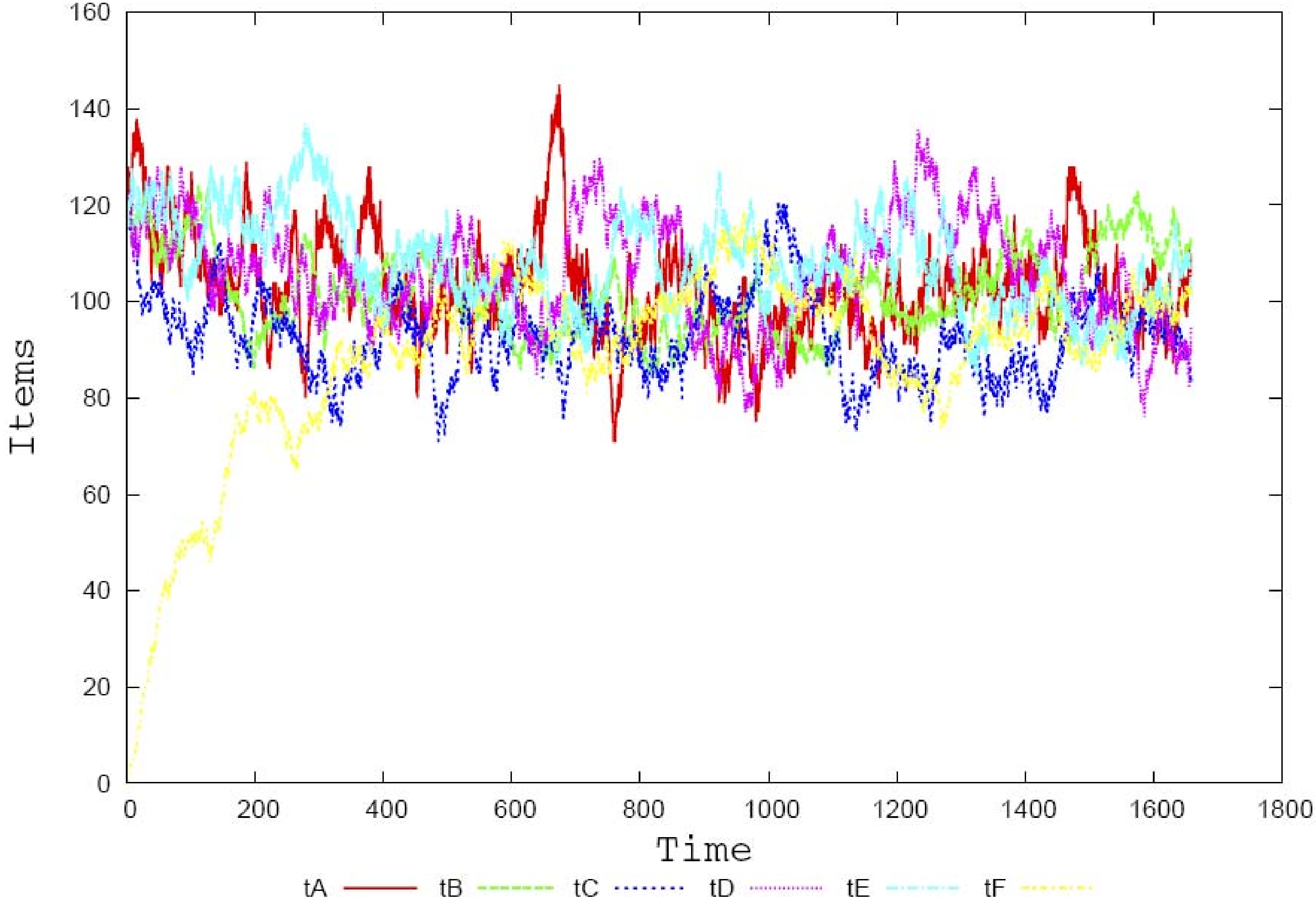
Simulation

- Starting from the previous model and the complete specification it is possible to run simulations
- It is necessary to provide parameters for temporal activities in the shape of rates
- The rates should reflect the deployment conditions, although at this step it is not so crucial

Simulation Results



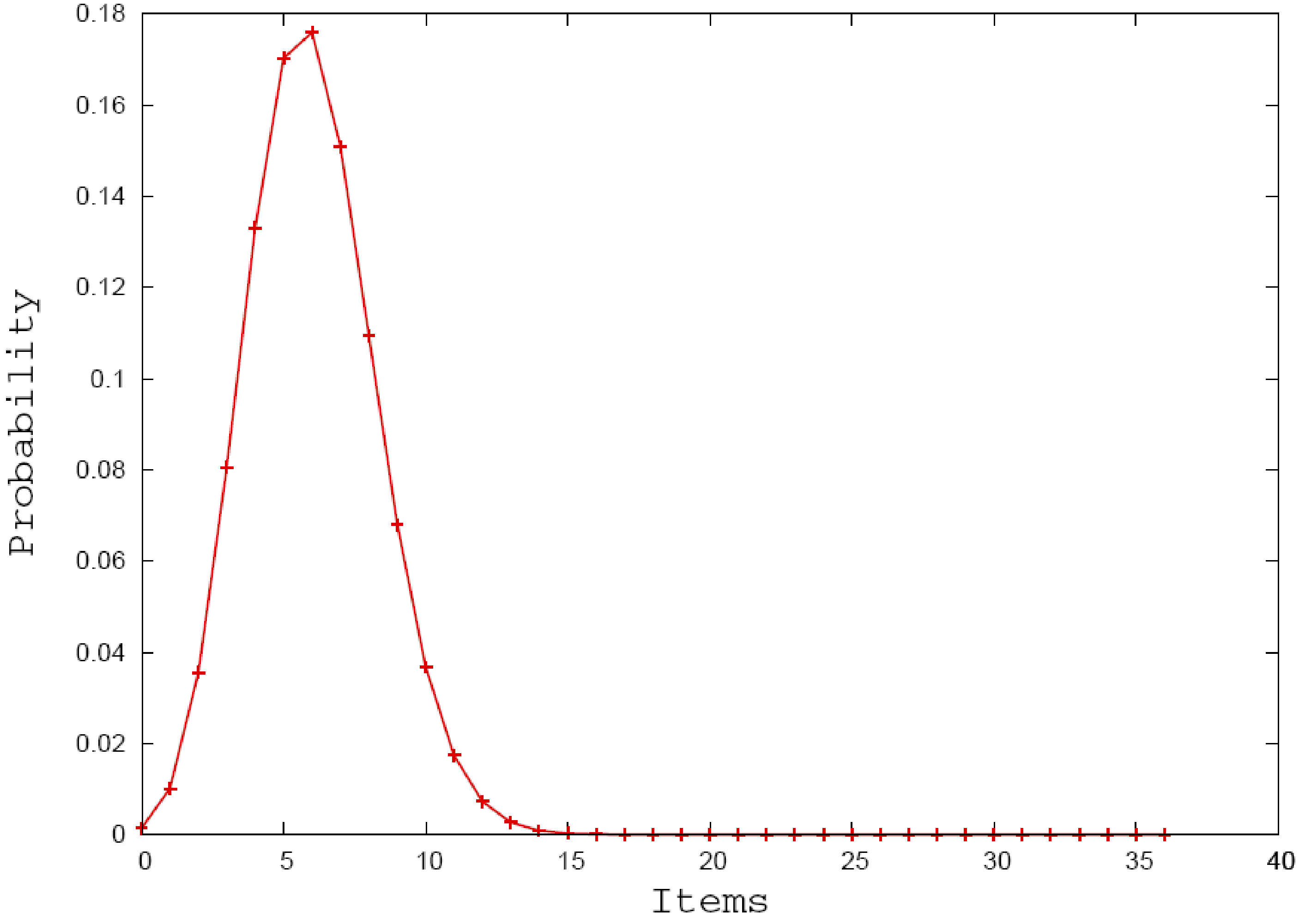
Simulation Results



Formal Verification

- Using the model checking capabilities of PRISM we can verify model properties
- For instance, we analyse the probabilistic distribution of artefact content
- The statement to verify is *“Which is the steady-state probability for the variable tA to assume the value Y?”*
- In PRISM syntax this is equivalent to an experiment with the formula *S=? [tA=Y]*

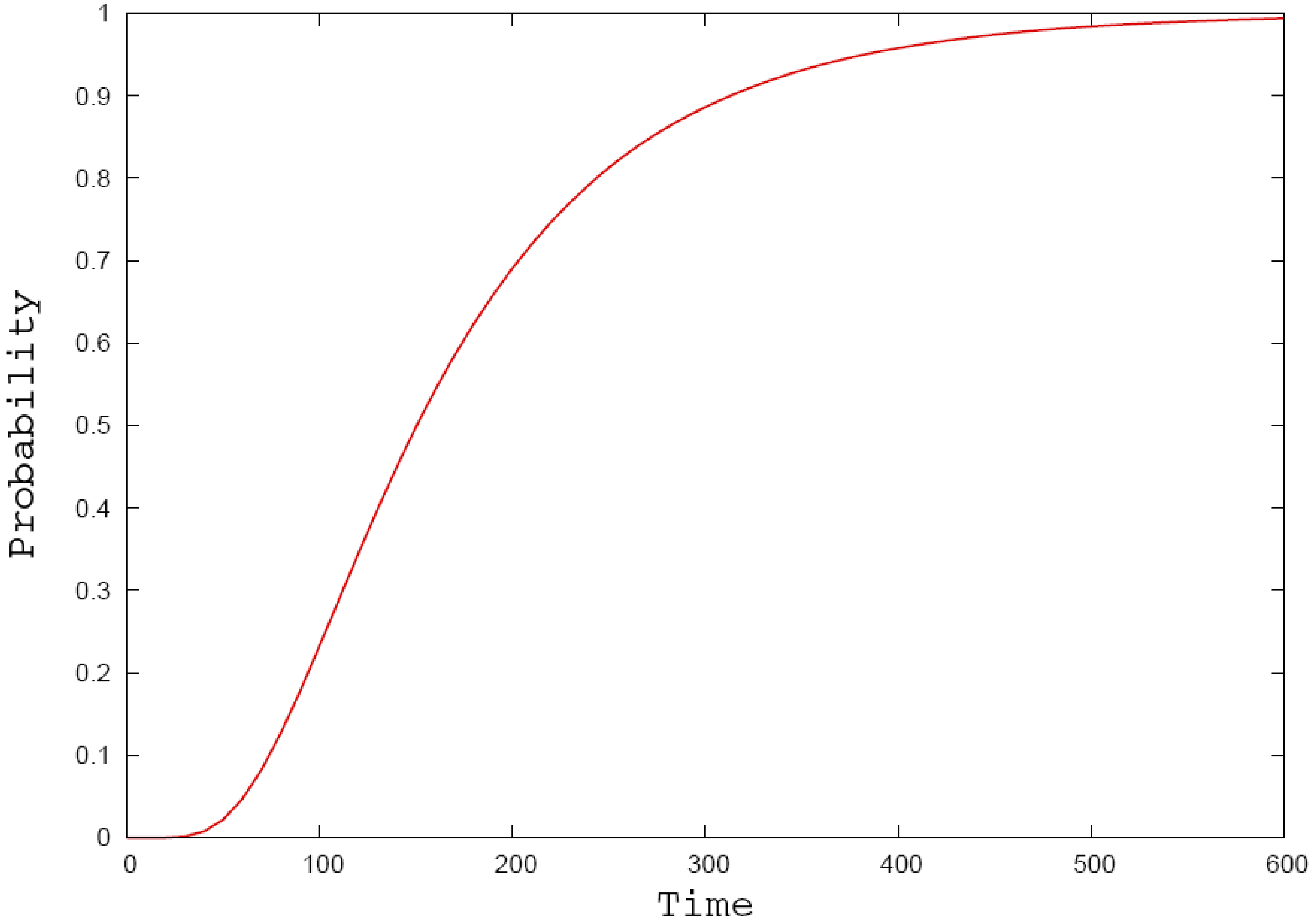
Model Checking Results



Formal Verification

- We want to analyse the behaviour of the system with respect to time
- This kind of analysis allows to evaluate system performance
- “Which is the probability for the node tB to be equal to 6 within Y time steps?“
- According to the PRISM syntax this is equivalent to the formula
  P=? [true U<=Y tB=6]

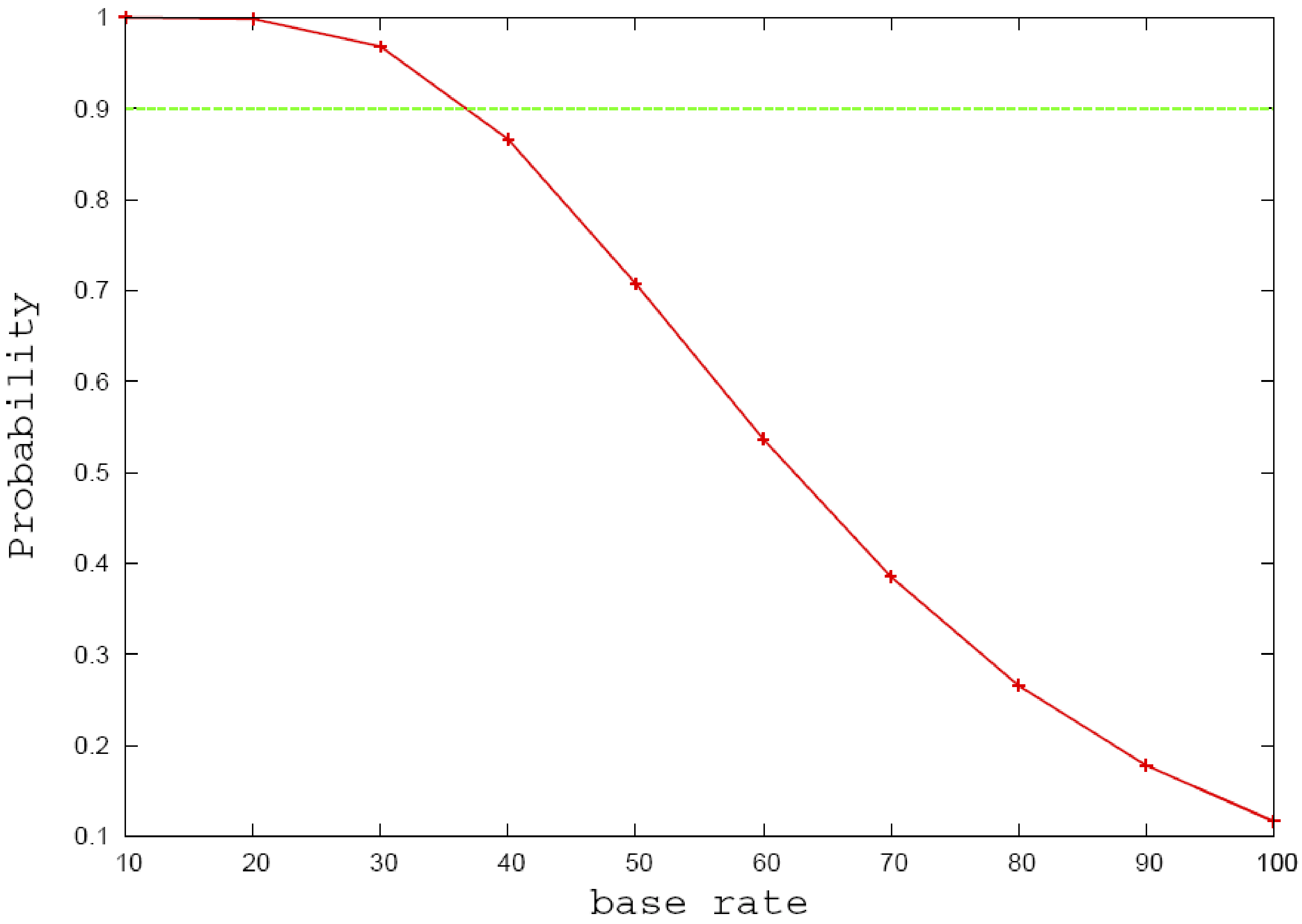
Model Checking Results



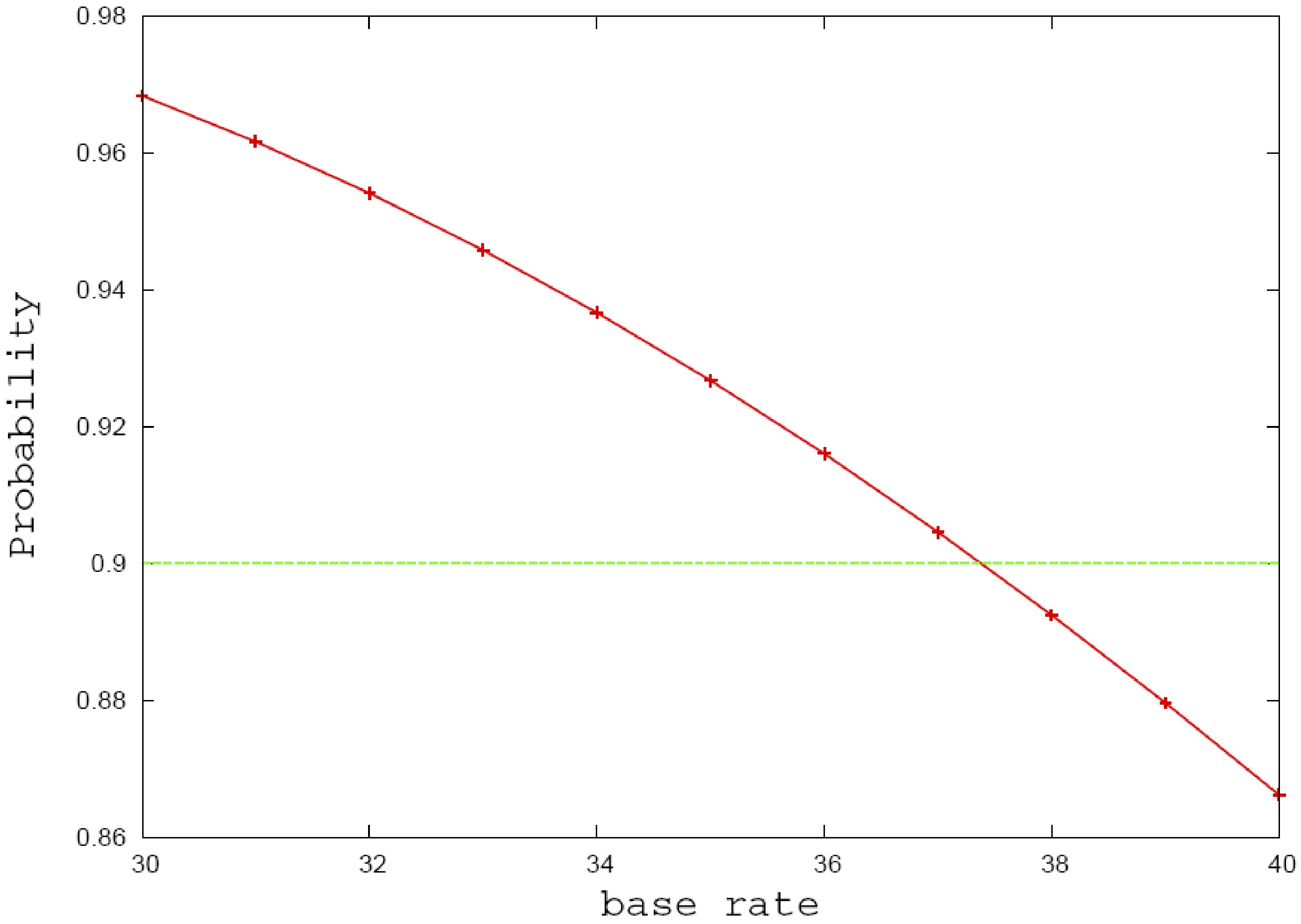
Tuning

- Since the qualititative dynamics of the system met the expectations, we now want to tune the system to achieve the desired performance
- We want to adjust the working rate in order for the system to reach the average value before a certain time
- We will perform analysis via model checking

Model Checking Results



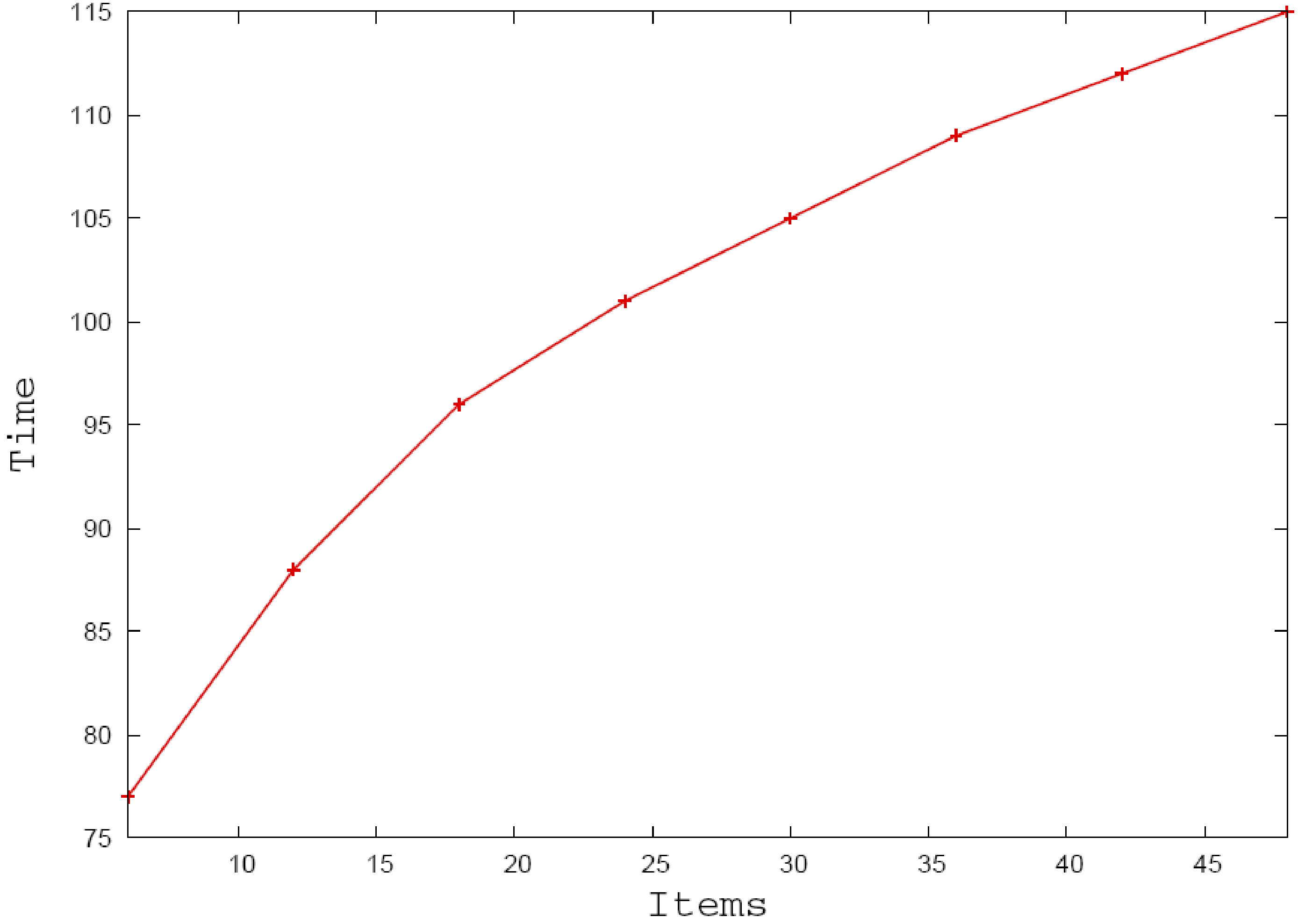
Model Checking Results



About Scalability

- We want now to consider the scalability of the strategy with respect to the number of items
- In particular we evaluate the time to reach the average value with respect to the number of items
- This analysis has been performed via model checking

About Scalability



Concluding Remarks

About Patterns

- We recognised a few simple patterns and proposed a pattern schema
- Simple patterns can be combined to produce more complex patterns: the results may be emergent!
- There may be incompatibilities among patterns
- Unfortunately, patterns by their very nature, tend to focus on structure rather than dynamics

About the Methodology

- The methodology has been applied to several case studies
- Other research groups are moving along a similar direction
- The methodology seems to be general enough to be applied to other domains other than software development
- In general, it is recognised that in the software development there should be more experimentation

About Formal Tools

- There are few uses of formal techniques and tools for analysing emergence
- There is a general belief that emergent systems are unformalizable, although it is true in some extent
- In our opinion the use of formal tools allowed us to gain a deeper insight in emergence and self-organisation
- The main limitation is related to the scale of the problem

About the Case Study

- We provided an alternative self-organising emergent solution to the Plain Diffusion problem
- The problem has been analysed according to our methodological approach
- Every step has been supported by the use of the PRISM tool

Publications

Publications: Overview

- All the activities related to the three years of PhD have been published in
  - Book Chapters: 1
  - Published in Journals: 3
  - Proceedings or Post-Proceedings published on Springer LNCS: 4
  - Technical Reports: 1
  - In Conferences or Workshops: 13

Publications: Book Chapters

- Luca Gardelli, Mirko Viroli, and Andrea Omicini. In Danny Weyns and Adeline M. Uhrmacher, editors, Agents, Simulation and Applications. *Simulation for the Development of Self-Organising Multi-Agent Systems*. Taylor & Francis, 2008. To Appear.

Publications: Technical Report

- Luca Gardelli. *Self-organization and coordination for multi-agent systems.* Technical report, European Science Foundation (ESF) MiNEMA Scientic Programme, November 2005. MiNEMA Exchange Grants Publications (no. 805)- Visit to Katholieke Universiteit Leuven.

Publications: In Journals

- Luca Gardelli, Mirko Viroli, Matteo Casadei, and Andrea Omicini. *Designing self-organising environments with agents and artifacts: A simulation-driven approach.* International Journal of Agent-Oriented Software Engineering (IJAOSE), 2(2), 2008, Inderscience. In Press.
- Matteo Casadei, Luca Gardelli, and Mirko Viroli. *Simulating emergent properties of coordination in Maude: the collective sort case.* Electronic Notes in Theoretical Computer Science (ENTCS), 175(2):59-80, June 21 2007, Elsevier. Proceedings of the Fifth International Workshop on the Foundations of Coordination Languages and Software Architectures (FOCLASA 2006).
- Mirko Viroli, Matteo Casadei, Luca Gardelli. *On the Collective Sort Problem for Distributed Tuple Spaces.* Science of Computer Programming (SCP), 2008, Elsevier. Invited Paper

Proceedings/Post-Proceedings on LNCS (1/2)

- Luca Gardelli, Mirko Viroli, and Andrea Omicini. *Design patterns for self-organising systems.* In Hans-Dieter Burkhard, Gabriela Lindemann, Rineke Verbrugge, and Laszlo Z. Varga, editors, Multi-Agent Systems and Applications V, volume 4696 of LNCS (LNAI), pages 123-132. Springer, Heidelberg, 2007. 5th International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2007, Leipzig, Germany, September 25-27,2007, Proceedings.
- Luca Gardelli, Mirko Viroli, Matteo Casadei, and Andrea Omicini. *Designing self-organising MAS environments: The collective sort case.* In Danny Weyns, H. Van Dyke Parunak, and Fabien Michel, editors, Environments for Multi-Agent Systems III, volume 4389 of LNCS (LNAI), pages 254-271. Springer, February 2007. 3rd International Workshop (E4MAS 2006), Hakodate, Japan, 8 May 2006. Selected Revised and Invited Papers.

Proceedings/Post-Proceedings on LNCS (2/2)

- Alessandro Ricci, Andrea Omicini, Mirko Viroli, Luca Gardelli, and Enrico Oliva. *Cognitive stigmergy: Towards a framework based on agents and artifacts*. In Danny Weyns, H. Van Dyke Parunak, and Fabien Michel, editors, Environments for Multi-Agent Systems III, volume 4389 of LNCS (LNAI), pages 124-140. Springer, February 2007. 3rd International Workshop (E4MAS 2006), Hakodate, Japan, 8 may 2006. Selected Revised and Invited Papers.
- Luca Gardelli, Mirko Viroli, and Andrea Omicini. *On the role of simulations in engineering self-organising MAS: The case of an intrusion detection system in TuCSoN.* In Sven A. Brueckner,Giovanna Di Marzo Serugendo, David Hales, and Franco Zambonelli, editors, Engineering Self-Organising Systems, volume 3910 of LNCS (LNAI), pages 153-166. Springer Berlin / Heidelberg, April 2006. Third International Workshop, ESOA 2005, Utrecht, The Netherlands, July 25, 2005, Revised Selected Papers.

Publications: Conferences & Workshops (1/5)

- Luca Gardelli, Mirko Viroli, and Andrea Omicini. *Design patterns for self-organizing multiagent systems.* In Tom De Wolf, Fabrice Sare, and Richard Anthony, editors, 2nd InternationalWorkshop on Engineering Emergence in Decentralised Autonomic System (EEDAS) 2007, pages 62-71, ICAC 2007, Jacksonville, Florida, USA, June 2007. CMS Press, University of Greenwich, London, UK.
- Mirko Viroli, Matteo Casadei, and Luca Gardelli. *A self-organising solution to the collective sort problem for distributed tuple spaces.* In 22th ACM Symposium on Applied Computing (SAC'07), pages 354-359, New York, NY, USA, March 2007. ACM Press. Special Track on Coordination Models, Languages and Applications.
- Matteo Casadei, Luca Gardelli, and Mirko Viroli. *A case of self-organising environment for MAS: the collective sort problem.* In Barbara Dunin-Keplicz, Andrea Omicini, and Julian Padget, editors, 4th European Workshop on Multi- Agent Systems (EUMAS 2006), volume 223, Lisbon, Portugal, December 2006. CEUR.

Publications: Conferences & Workshops (2/5)

- Matteo Casadei, Luca Gardelli, and Mirko Viroli. *Collective sorting tuple spaces.* In Andrea Omicini Flavio De Paoli, Antonella Di Stefano and Corrado Santoro, editors, Dagli oggetti agli agenti: Sistemi Grid, P2P e Self-*, AI*IA/TABOO Joint Workshop (WOA 2006), pages 173-180, Catania - Italy, September 2006. Technical University of Aachen.
- Matteo Casadei, Luca Gardelli, and Mirko Viroli. *Simulating emergent properties of coordination in Maude: the collective sorting case.* In Carlos Canal and Mirko Viroli, editors, 5th International Workshop on the Foundations of Coordination Languages and Software Architectures (FOCLASA), pages 57-75, CONCUR 2006, Bonn, Germany, August 2006. University of Malaga, Spain.
- Luca Gardelli, Mirko Viroli, and Matteo Casadei. *On engineering self-organizing environments: Stochastic methods for dynamic resource allocation.* In Danny Weyns, H. Van Dyke Parunak, and Fabien Michel, editors, 3rd International Workshop on Environments for Multi-Agent Systems (E4MAS 2006), pages 96-101, AAMAS 2006, Hakodate, Japan, May 2006.

Publications: Conferences & Workshops (3/5)

- Alessandro Ricci, Andrea Omicini, Mirko Viroli, Luca Gardelli, and Enrico Oliva. *Cognitive Stigmergy: A Framework Based on Agents and Artifacts.* In Danny Weyns, H. Van Dyke Parunak, and Fabien Michel, editors, 3rd International Workshop on Environments for Multi-Agent Systems (E4MAS 2006), pages 44-60, AAMAS 2006, Hakodate, Japan, May 2006.
- Luca Gardelli, Mirko Viroli, and Matteo Casadei. *Engineering the environment of self-organising multi-agent systems exploiting formal analysis tools.* In Congresso AICA 2006, Cesena - Italy,September 2006. AICA – Associazione Italiana per l'Informatica e il Calcolo Automatico.
- Luca Gardelli, Mirko Viroli, and Andrea Omicini. *Exploring the dynamics of self-organising systems with stochastic -calculus: Detecting abnormal behaviour in MAS.* In Robert Trappl, editor, Cybernetics and Systems 2006, volume 2, pages 539-544, Vienna, Austria, April 2006. Austrian Society for Cybernetic Studies. 18th European Meeting on Cybernetics and Systems Research (EMCSR 2006), 5th International Symposium From Agent Theory to Theory Implementation (AT2AI-5).

Publications: Conferences & Workshops (4/5)

- Alessandro Ricci, Andrea Omicini, Mirko Viroli, Luca Gardelli, and Enrico Oliva. *Cognitive Stigmergy: A Framework Based on Agents and Artifacts.* In Marie-Pierre Gleizes, Gal A. Kaminka, Ann Nowe, Sascha Ossowski, Karl Tuyls, and Katja Verbeeck, editors, 3rd European Workshop on Multi-Agent Systems (EUMAS 2005), pages 332-343, Brussels, Belgium, December 2005. Koninklijke Vlaamse Academie van Belie voor Wetenschappen en Kunsten.
- Luca Gardelli, Mirko Viroli, and Andrea Omicini. *On the role of simulation in the engineering of self-organising systems: Detecting abnormal behaviour in MAS.* In Flavio Corradini, Flavio De Paoli, Emanuela Merelli, and Andrea Omicini, editors, AI*IA/TABOO Joint Workshop Dagli oggetti agli agenti: simulazione e analisi formale di sistemi complessi (WOA 2005), pages 85-90, Camerino, MC, Italy, November 2005. Pitagora Editrice Bologna.

Publications: Conferences & Workshops (5/5)

- Luca Gardelli, Mirko Viroli, and Andrea Omicini. *On the role of simulations in engineering self-organizing MAS: the case of an intrusion detection system in TuCSoN.* In Sven Brueckner, Giovanna Di Marzo Serugendo, David Hales, and Franco Zambonelli, editors, 3rd International Workshop on Engineering Self-Organising Applications (ESOA 2005), pages 161-175, AAMAS 2005, Utrecht, The Netherlands, July 2005.
- Luca Gardelli, Mirko Viroli, and Andrea Omicini. *Engineering self-organizing MAS with coordination artifacts and ACCs.* In Jean-Marie Jacquet and Gian Pietro Picco, editors, 7th International Conference on Coordination Languages and Models (COORDINATION 2005), Namur, Belgium, April 2005. Poster.