Self-* and multi-agent systems

Applications to collective robotics and problem solving

Gauthier Picard

SMA / G2I / École Nationale Supérieure des Mines de Saint-Étienne (ENSM.SE)

picard@emse.fr

November 6, 2008

Contents

Motivations

- Technical motivations
  - More powerful computers (Moore's law)
  - Systems become more open (Internet, Ambient, etc.)
  - Interdependencies, non linearities between them
- Theoretical consequences and limitations
  - Incompleteness theorems of Gödel
  - Law of requisite variety of Ashby
  - "No free lunch theorems" of Wolpert and Mac Ready

$\Rightarrow$ Designing and manipulate *Complex Systems*

Motivations

- Technical motivations
  - More powerful computers (Moore's law)
  - Systems become more open (Internet, Ambient, etc.)
  - Interdependencies, non linearities between them
- Theoretical consequences and limitations
  - Incompleteness theorems of Gödel
  - Law of requisite variety of Ashby
  - "No free lunch theorems" of Wolpert and Mac Ready

$\Rightarrow$ Designing and manipulate *Complex Systems*

Emergence technologies for CS

- ✗ We cannot formally prove that sufficiently “complex” software are safe
- ✗ Systems openness goes against their *a priori* checking

⇒ Standard design is today inappropriate

⇒ New systems must be self-* (self-control, self-repair, self-healing, self-organisation, etc.)

⇒ New theories and methods for designing emergent functionality systems

Emergence technologies for CS

- ✗ We cannot formally prove that sufficiently “complex” software are safe
- ✗ Systems openness goes against their *a priori* checking

⇒ Standard design is today inappropriate

⇒ New systems must be self-* (self-control, self-repair, self-healing, self-organisation, etc.)

⇒ New theories and methods for designing emergent functionality systems

Contents

Autonomic and Self-*
Illustration

http://ants.gsfc.nasa.gov/

Autonomic and Self-*

Demonstration

http://ants.gsfc.nasa.gov/

Autonomic Computing

- Initiative started by IBM in 2001
- Autonomy-oriented computation, Emergent computation
- Aims at developing self-managed systems
- Inspired from social and natural systems
  - Ant, termite colonies, bees, wasps, etc.
  - Human societies
  - etc.

⇒ *Collective* actions of the sub-parts (**agents**) lead to the whole system *adaptation*

Autonomic Computing (cont.)

Self-* Properties

- Self-tuning: Automatic adjustment of internal parameters
- Self-configuration: Automatic configuration of components
- Self-healing: Automatic discovery, and correction of faults
- Self-optimization: Automatic monitoring and control of resources to ensure the optimal functioning with respect to the defined requirements
- Self-protection: Proactive identification and protection from arbitrary attacks
- ...

From Self-* to Self-Organisation

- Topology of an organisation determines the functionality of the system at a given time
- Changing the organisation → changing the function

⇒ A *self-organising system* is a system able to change its internal organisation as to reach an equilibrium (adaptation)

- This environment is the engine for stability and convergence (ex: stigmergy)

From Self-* to Self-Organisation

- Topology of an organisation determines the functionality of the system at a given time
- Changing the organisation $\rightarrow$ changing the function

$\Rightarrow$ A *self-organising system* is a system able to change its internal organisation as to reach an equilibrium (adaptation)

- This environment is the engine for stability and convergence (ex: stigmergy)

From Self-* to Self-Organisation

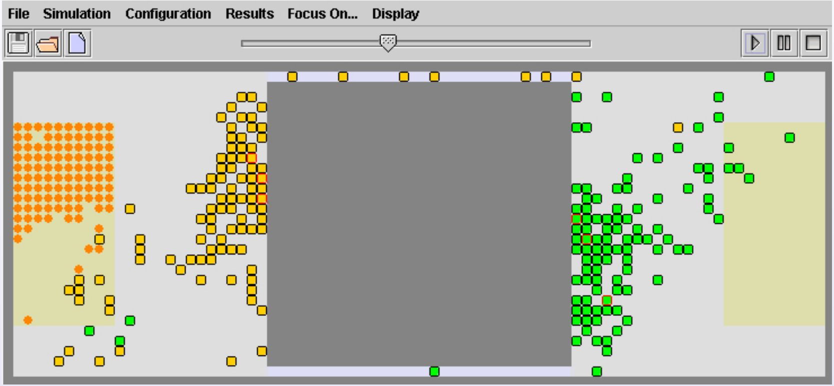
- Topology of an organisation determines the functionality of the system at a given time
- Changing the organisation $\rightarrow$ changing the function

$\Rightarrow$ A *self-organising system* is a system able to change its internal organisation as to reach an equilibrium (adaptation)

- This environment is the engine for stability and convergence (ex: stigmergy)

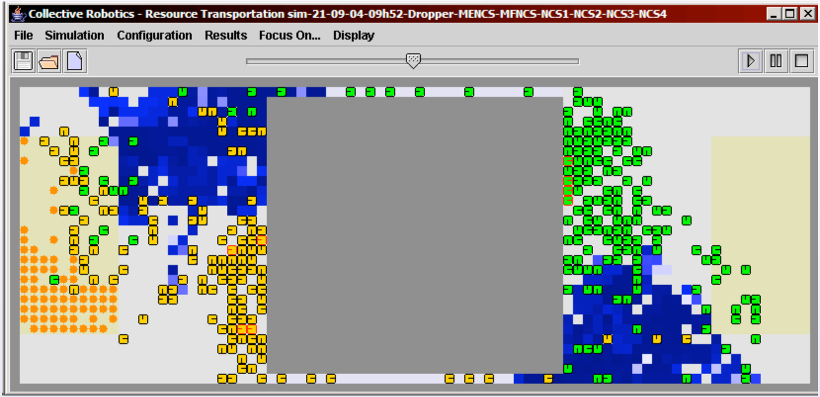
Contents

Collective Robotics



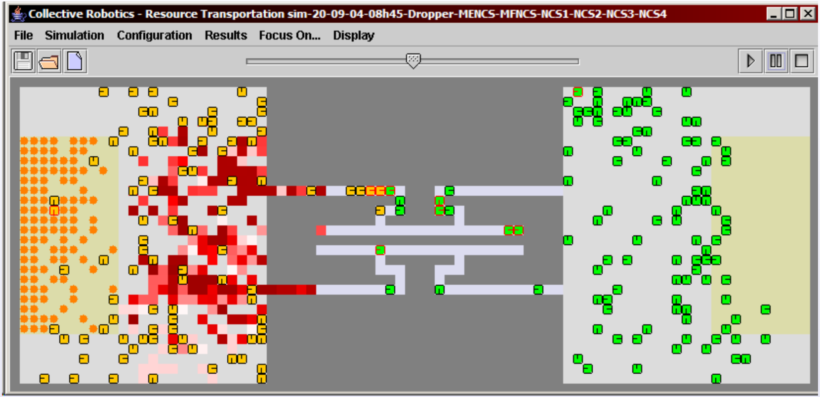
[Picard and Gleizes, 2005]

Virtual Pheromones



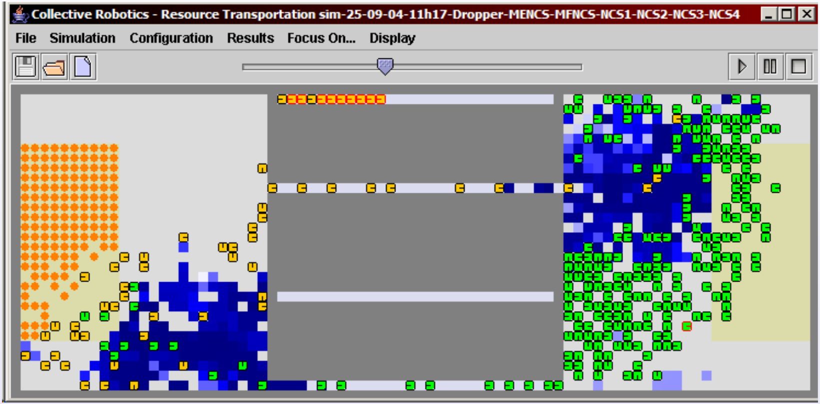
Virtual markers positions for all robots and 2 goals

Virtual Pheromones (cont.)



Virtual markers positions for all robots and for the goal *reach the drop zone*

Virtual Pheromones (cont.)



Virtual markers positions for all robots in a dynamical environment

Contents

Problem Solving

- Self-organisation is also a way to solve problems
  - distribution
  - decentralisation
  - dynamics

  $\Rightarrow$ Distributed Constraint Satisfaction/Optimisation
- Examples:
  - Ant colony optimisation [Bonabeau et al., 1999]
  - Particle swarm optimisation
  - ...
- Applications
  - Scheduling:
    - Course scheduling [Picard et al., 2005]
    - Manufacturing control [Clair et al., 2008]
  - Frequency assignment [Picard et al., 2007]
  - ...

Problem Solving

- Self-organisation is also a way to solve problems
  - distribution
  - decentralisation
  - dynamics

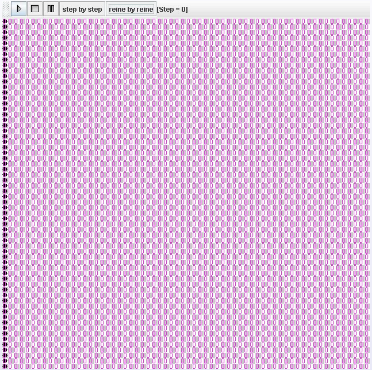
  ⇒ Distributed Constraint Satisfaction/Optimisation
- Examples:
  - Ant colony optimisation [Bonabeau et al., 1999]
  - Particle swarm optimisation
  - ...
- Applications
  - Scheduling:
    - Course scheduling [Picard et al., 2005]
    - Manufacturing control [Clair et al., 2008]
  - Frequency assignment [Picard et al., 2007]
  - ...

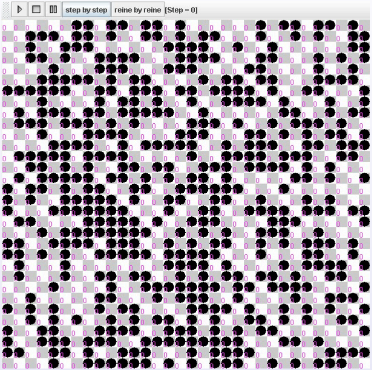
Problem Solving

- Self-organisation is also a way to solve problems
  - distribution
  - decentralisation
  - dynamics

  ⇒ Distributed Constraint Satisfaction/Optimisation
- Examples:
  - Ant colony optimisation [Bonabeau et al., 1999]
  - Particle swarm optimisation
  - ...
- Applications
  - Scheduling:
    - Course scheduling [Picard et al., 2005]
    - Manufacturing control [Clair et al., 2008]
  - Frequency assignment [Picard et al., 2007]
  - ...

Idea





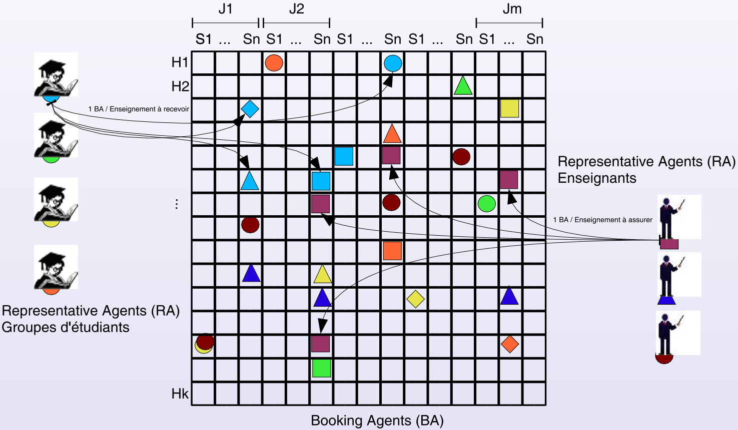
[Picard and Glize, 2006]

Course Scheduling [Picard et al., 2005]

- Dynamic solver for university schedule
- Constraints can be updated at run-time
- Schedules are iteratively built

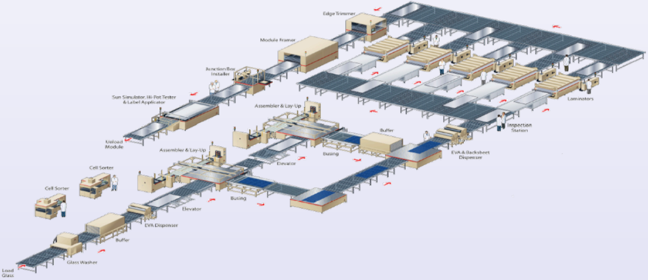
Course Scheduling [Picard et al., 2005]

- Dynamic solver for university schedule
- Constraints can be updated at run-time
- Schedules are iteratively built



Manufacturing Control [Clair et al., 2008]

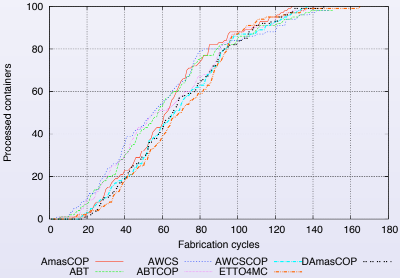
- Increasing decentralisation to decrease the impact of perturbations
- Machines are agentified are cooperate to find an optimal task assignment and to respond to commands
- Benchmark of the Colline group to compare self-organising approaches



Manufacturing Control [Clair et al., 2008] (cont.)

[Clair et al., 2008]

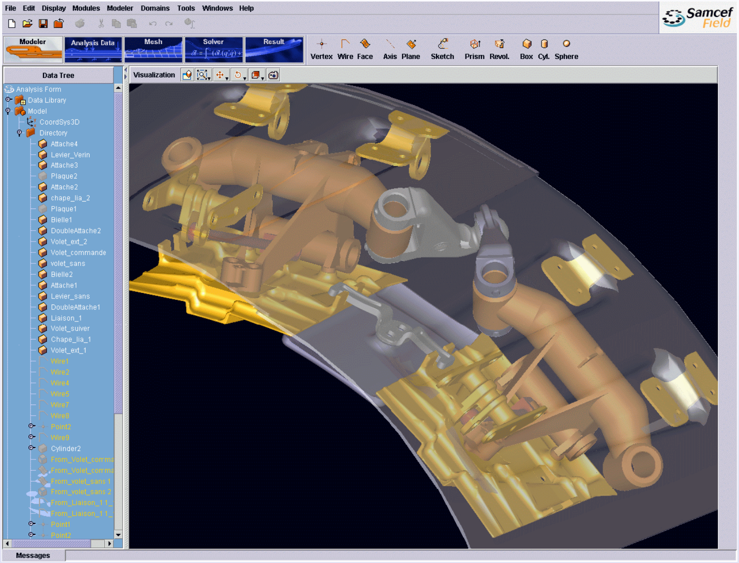
- ✓ cooperative behaviour leads to better results for unknown commands
- ✗ more messages than classical distributed approaches (ABT, AWCS)



Contents

Mechanical Design

FP5 European Project SYNAMEC



Mechanical Design [Capera et al., 2004, 2005]

FP5 European Project SYNAMEC

Example of self-assembly and self-tuning

Contents

Conclusions

- Variety of inspirations
- *Multi-agent* based methodologies to design *adaptive systems*
  - Bottom-up approach
    - Environment modelling
    - Agent design
    - Self-* rules specification
  - Distributed solving/optimisation
    - ✓ Adaptation to dynamics
    - ✓ Problem trace
    - ✓ Anytime
- Simulation / Modelling
- Numerous application fields
- Projects
  - Ambient intelligence (ex: crisis management)
  - Multi-disciplinary optimisation
  - Adaptive manufacturing and coordination
  - ...

Conclusions (cont.)

Opening-up to Health Domain

- Scheduling and resource allocation for medical staff
  - Adaptation to dynamics
  - Encapsulation of data and locality principle → privacy
- Simulation of complex systems
  - Hospital viewed as an *ecosystem*
  - Computational Biology
    - *e.g.* microMega project
    - *e.g.* Paul Bourgine's works
  - Molecule (Un)folding
- Collective robotics
  - *e.g.* as in iWARD project
- Ambient Intelligence
  - Infrastructure to help medical staff in crisis management
  - *e.g.* AmICriM project
- ...

Questions ?

References

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

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.

D. Capera, G. Picard, M.-P. Gleizes, and P. Glize. A Sample Application of ADELFE Focusing on Analysis and Design : The Mechanism Design Problem. In M.-P. Gleizes, A. Omicini, and F. Zambonelli, editors, *Fifth International Workshop on Engineering Societies in the Agents World (ESAW'04), 20-22 October 2004, Toulouse, France*, volume 3451 of *Lecture Notes in Artificial Intellignece (LNAI)*, pages 231–244. Springer-Verlag, 2005.

G. Clair, M.-P. Gleizes, E. Kaddoum, and G. Picard. Self-Regulation in Self-Organising Multi-Agent Systems for Adaptive and Intelligent Manufacturing Control. In *Second IEEE International Conference on Self-Adaption and Self-Organization (SASO 2008), Venice, Italy, 20-24 October 2008*. IEEE Computer Society, 2008.

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.

G. Picard, C. Bernon, and M.-P. Gleizes. Emergent Timetabling Organization. In *Multi-Agent Systems and Applications IV - 4th International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS'05), 15-17 September 2005, Budapest, Hungary*, volume 3690 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 440–449. Springer-Verlag, 2005.

G. Picard, M.-P. Gleizes, and P. Glize. Distributed Frequency Assignment Using Cooperative Self-Organization. In *First IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'07), Boston, Mass., USA, July 9-11, 2007*, pages 183–192. IEEE Computer Society, 2007.