Institut de Recherche
en Informatique de Toulouse

Rapport interne
IRIT/2005-18-R

# Self-Organisation in Multi-Agent System

Editors
Di Marzo Serugendo Giovanna
Gleizes Marie-Pierre
Karageorgos Anthony

UPS - IRIT 31062 Toulouse cedex 9 05 61 55 67 65 info@irit.fr www.irit.fr

**List of Authors:**

Bernon Carole, IRIT - Toulouse – France,
Besse-Patin Camille, IRIT- Toulouse – France,
Bongard Markus, University Miguel Hernandez, Alicante, Spain,
Bourjot Christine, LORIA – Nancy – France,
Camps Valérie, IRIT- Toulouse – France,
Capera Davy, IRIT- Toulouse – France,
Chevrier Vincent, LORIA – Nancy – France,
Deguet Joris, LEIBNIZ Laboratory, France,
Di Marzo Serugendo Giovanna, University of Geneva - Geneva – Switzerland,
Fernandez Eduardo, University Miguel Hernandez, Alicante, Spain,
Gaud Nicolas, University of Technology of Belfort-Montbeliard - Belfort – France,
Gleizes Marie-Pierre, IRIT- Toulouse – France,
Glize Pierre, IRIT- Toulouse – France,
Hassas Salima, LIRIS - Université Claude Bernard-Lyon 1 – Lyon – France,
Hilaire Vincent, University of Technology of Belfort-Montbeliard - Belfort – France,
Holvoet Tom, KU.Leuven, Leuven – Belgium,
Karageorgos Anthony, University of Thessaly, Volos – Greece,
Lopardo Gabriel, Univniversity of Girona, Spain,
Mano Jean-Pierre, IRIT- Toulouse – France,
Rodriguez Sebastian, University of Technology of Belfort-Montbeliard - Belfort – France,
Valckenaers Paul, K.U.Leuven, Leuven, Belgium,
Verstraete Paul, K.U.Leuven, Leuven – Belgium,
Vouros George, University of Agean, Greece.

# Table of contents

# 1. Introduction

The aim of the Self-Organisation (SELF-ORG) Agentlink III TFG is to work on the issue of self-organisation in complex distributed systems, and in particular Multi-Agent Systems

(MAS). To obtain a broader view, the aim is to study agent systems differing adequately in terms of granularity, cardinality and heterogeneity. Particular interest is on approaches that can be used to design MAS capable of exhibiting self-organising behaviour. The work concentrates on attempting to answer questions of major interest such as:

- How could self-organisation mechanisms already applied in other disciplines be reused in the design of MAS?
- How could we derive appropriate self-organisation mechanisms for MAS starting from a set of application requirements?
- Can a set of criteria suitable for characterisation of self-organisation and emergence in MAS be defined?
- How could we compare the different mechanism of self-organisation? Could we define suitable measures and comparison criteria?
- How could we strengthen the links with other disciplines such as biology, systemics, sociology and physics? Due to the diversity of the issues involved with self-organisation, participation of people from different disciplines such as biology, systemic, sociology, physics and software engineering was strongly encouraged in an attempt to highlight problem of self-organisation from different perspectives.

## 2. Summary of the second TFG meeting[1]

A kick-off meeting of the SELF-ORG TFG took place in Rome in July 2004 which defined the context and carried out initial steps of the collaborative work. The second meeting was held in Ljubljana, Slovenia. The particular objectives of the second meeting were:

- To provide presentations from active participants of the way they apply self-organisation in the different case studies selcted elaborated after the previous meeting in Rome.
- To define a number of measures and assessment criteria for self-organisation.
- To derive definitions of several notions with an interdisciplinary view such as: emergence, complexity, adaptive systems, and to list their commonalities and their differences, as well as the theories underlying them.
- To bring in touch people from all the computing science research communities working on self-organisation and emergent behaviour such as MAS, autonomic computing, pervasive computing, grid computing and distributed computing
- To establish and enhance links with other disciplines through contacts with networks such as EXYSTENCE, or researcher communities involved in the new EU NEST-PATHFINDER initiative (Tackling Complexity of Science).
- To carry our preparatory work towards the publication of a book gathering and enhancing the results of the SELF-TFG collaborative efforts.

An important outcome of the first meeting in Rome was the decision to establish case studies capable of acting both as benchmarks and as working examples faciliating the understanding of self-organisation, emergent properties and their underlying mechanisms. Therefore, half of the meeting of the SELF-ORG TFG in Ljubljana was dedicated to the work on the case studies which was carried out between the two meetings. This involved describing the selected case studies using a uniform framework and proposing agent-based solutions for modelling and/or for implementing them. The first case study is related to computer networks management. It was proposed by Salima Hassas from Lyon University (France). Four approaches used to treat it, based on holonic systems, stigmergy mechanisms, peer to peer communications and routing and delegation respectively, were presented. Paul Valckenaers from KU Leuven (Belgium) proposed the second case study which concerns manufacturing control. In that case study two self-organising mechanisms were applied: ant algorithms and cooperative agents. Finally, the third case study concerned simulating space conformation of molecules and was was introduced by Pierre Glize from IRIT (France). The problem posed in the third case study was addressed by using self-organisation arising from cooperative agents.

Since a major aim of the SELF-ORG TFG is the establishment of links with other disciplines, the second part of the Ljubljana meeting was devoted to an invited talk presented by Markus Bongard from University Miguel Hernandez, Alicante, Spain. The talk was in the field of neuro-biology and it concerned the complexity in mammalian visual systems. During the presentation, it was made clear that there is a large potential for using agents in other disciplines, for instance in simulating the behaviour of the brain neurons with agents. Establishing links with other disciplines offers two major advantages: on the one hand, it enables finding new application domains where self-organisation mechanisms could be applied and on the other it facilitates the transfer of self-organising mechanisms applied in other areas to computing applications.

The rest of the meeting was devoted to the definition of “emergence”. Three aspects of the concept of emergence were examined. Firstly, a number of necessary *characteristics* that an emergent phenomenon must show were highlighted. Subsequently, the *properties* an artificial system must have to produce emergent phenomena were listed. Finally, the *operational characteristics*, that must be present in systems in order for their behaviour to qualify its result as emergent were enumerated. Although much progress has been in this direction by identifying necessary properties, essential criteria and operational requirements, additional work is still needed to finalise the definition efforts.

# 3. Main Concepts Studied

## 3.1. Introduction[1]

The systems we are interested in are composed of several elements which collectively produce “something" at global level; it is not chaos, but an *organisation* (a pattern, a process to produce this pattern and a function: the system fulfils some requirements). We are interested in the study, analysis and design of systems capable of producing a collective outcome which would result from local interactions between its simple individual components. More precisely, what is interesting is the ability of such a system to have a collective response such as its complexity is out the individual simplicity. The terms involved, for example organisation, self-organisation and emergence, are been used in various disciplines, such as mathematics, physics, biology, and philosophy. These terms are increasingly used in computer science and in particular they are widely met in the MAS research community. Wce need to establish a common, consensual and operational meaning of them to enable researchers to know if their artificial systems are self-organising or not, if there is emergence or not. This leads the TFG SO group to work in a first time on the definition of the two main concepts such as self-organisation and emergence.

---

[1] Authors : Di Marzo Serugendo G., Gleizes MP., Karageorgos A.

## 3.2. Self-Organisation[2]

### 4.2.1. Definitions

Self-organisation is an attractive way to handle the dynamic requirements in software. It refers to a process where a system changes its internal organisation to adapt to changes in its goals and the environment without explicit external control. Self-organisation often results in emergent behaviour that can be either desirable or undesirable. Due to the dynamism and openness of contemporary agent environments and the ever increasing distribution, complexity and dynamic changes in application requirements, understanding the mechanisms that can be used to model, assess and engineer self-organising behaviour in MAS is an issue of major interest.

> We define **self-organisation** as the mechanism or the process enabling a system to change its organisation without explicit external command during its execution time.

It is also necessary to distinguish between systems where there is no internal and external explicit control from those where there is an internal centralized control. For example, in a termites' society, the different arches are all located at the same distance from the queen due to a pheromone gradient. The queen broadcasts this information and this action is an internal control. In other way, the different arches are built without internal or external control. By consequence the following definitions have been given:

> **Strong self-organising systems** are those systems where there is no explicit central control either internal or external.
>
> **Weak self-organising systems** are those systems where, from an internal point of view, there is re-organisation maybe under an internal (central) control or planning. This kind of systems can be illustrated by the example of the termites which put the bullet in a circle under the control of the queen.
>
> **A self-organised system** is a system which exhibits a new organisation due to a self-organisation process.
>
> **A self-organising system** is simply a system which is in the process of exhibiting internally at least some reorganisation mechanisms consistent with the self-organisation definition.

### 4.2.2. Properties of Self-Organising Behaviour

- *Absence of external control* (autonomy)
- *Decentralised control:*
  not mandatory since we can have an internal centralised control;
  mandatory in MAS. There is no central authority or centralisation in information flow. It is a consequence of the locality of interactions and of simple rules that limit access to global information.

---

[2] Authors : Bourjot C., Chevrier V., Di Marzo Serugendo G., Gleizes MP., Karageorgos A.

- *Dynamic operation* (time evolution):
  Since the organisation evolves without external control, it implies a continuous dynamic process.
- *Fluctuations (noise/searches through options) property not relevant to SO*
- *Symmetry breaking (loss of freedom/heterogeneity)*

### 4.2.3. Characteristics of Self-Organising Systems

- *Global order endogenous*
- *Emergent properties arise from local interactions*:
  Properties that can be observed at global level and that can not be deduced by simply observing individual behaviours. See Emergence.
- *Dissipation* (energy usage/far-from-equilibrium):
  In absence of external perturbation, the system is expected to stabilise in some states in which emergent properties can be observed. This implies a kind of dissipation of some "energy" (otherwise the system would be continuously changing).
- *Instability* (self-reinforcing choices/nonlinearity), *parameter sensibility*:
  These systems are characterized mainly by non linear dynamics (small fluctuations when near some critical point can provide significant modifications of the system), by sensibility to initial conditions and parameter sensibility (small changes on a parameter produce different patterns). Thus the overall properties cannot be understood simply by examining separately the components (see complexity)
- *Multiple equilibriums* (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):
  Complexity comes from the irreducibility of the global properties to a combination of local behaviours.
- *Hierarchies* (multiple nested self-organised levels)
- *Simple local rules*:
  Simple individual behavioural rules lead to complex patterns. The information stored in rules description is less than the information needed to describe the pattern: it is only needed to store information about way to produce the pattern, not the pattern itself.

### 4.2.4. Mandatory Requirements

- Absence of external control (autonomy) is implicit (needed in general, and in MAS)
- Dynamic operation (time evolution) is needed
- Global order endogenous is needed

### 4.2.5. Operational Aspects (Requirements) for Self-Organisation in MAS

- Self-organizing systems imply emergence
- Several agents – trivial unless not MAS
- Many local interactions
- Perceptions by agent: perception of other agents and of global result
- Own perception (not necessary partial perception)
- Local rules at the agent level
- Interaction between the system and the environment when they can be differentiated.
- Two kinds of systems: those that are differentiated from the environment, those that are combined with the system (called eco-system). We need also to differentiate between the environment of agents, and the environment of MAS. True also in other systems.
- Decentralized internal control
- Dynamic environment
- Environmental constraints to make the system converge
- Many interaction or many events.

### 4.2.6. Measures for Self-Organisation

To the best of our knowledge, there are currently no works on the measures specific to self-organisation. Here is a first attempt to provide some criteria or quantitative way of measuring self-organising systems in order to compare self-organising systems with each other and with traditional approaches.

Measurements can be viewed from three points of view according to organisation side:

- A structure that can be observed,
- A process that produces and maintains that structure,
- A function: a self-organising process, at least an artificial one, is expected to fulfil a certain purpose.

Furthermore, a self-organising system can be studied from the local and the global perspectives (or more if we distinguish multiple nested self-organised levels). Measurements can be undertaken along these three aspects at each level.

Measurements from the function point of view can be related to the problem of how the SO system is able to fulfil its purpose: measures are done according to characteristics of the problem being solved by the system. Such measures are similar to those done in classical systems.

Since self-organising systems are dynamical ones that stabilise, the process and structure based measures are more interesting. Process based measurement is related to the dynamics (evolution over the time) of the systems, since structure based measurement is related to the observation of the structure (when stabilised).

Here are some specific measures:

- Capacity to reach an organisation, able to fulfil the goal of the system as a whole, once the system is started (success/failure/time required, convergence)

- Capacity to reach a re-organisation after a perturbing event (success/failure/time required)
- Degree of decentralised control (central/totally decentralised/hybrid)
- Capacity to resist or not to perturbations: stability/adaptability.

## 3.3. Emergence[3]

### 4.3.1. What Emerge?

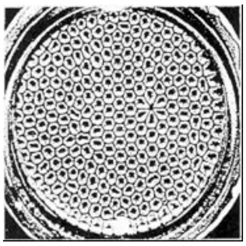
What is emergence anyway? One starting point for finding an answer to this question are dictionary definitions. After consulting several physical as well as on-line dictionaries, it is obvious that no definition can be found which allows us to enumerate a list of criteria for identifying emergence. The verb "to emerge" means no more or no less than "to become apparent" or "to come out into view". The object of the emergence is often called phenomenon and can be a structure or a framework, or behaviour, or a function (not in the sense mathematic function but in the sense functionality of a system).

What can be extracted from these "synonyms" is one thing: emergence is all about the **observation** of a phenomenon, a process, a system, an activity; in other words, emergence is "in the eye of the beholder". Any phenomenon that arises from a system cannot be recognized as an emergent phenomenon unless there is an observer (either a human or a system) for who the phenomenon has a meaning. In fact, a phenomenon does not exist without its relationship to the observer.

Let us illustrate this by a number of examples. In complex systems, the research very often centers on the emergent global dynamics of a whole system. It is usual in this approach to view the global properties of the system as emerging from the actions of its parts, rather than seeing the actions of the parts as being imposed from a dominant central source [30][31]).

The concept of the *shortest path* between an ant nest and a food source exists only for who observes that density of ants and pheromones in the biological environment of the ants.

In the Bénard’s cells, the observer sees a structure emerges. The warmth is transmitted by a regular flux from the bottom towards the surface by conduction. When we continue to warm up, we leave this equilibrium state towards a state far from this equilibrium state due to the temperature difference between the two regions. This phenomenon leads to the progressive apparition to an observer of chaotic circulation of the fluid which grows up until the boiling point. But before the boiling point, there is a critical point, where there is not a great circulation running to enable quick warmth dispersal inside the fluid. A self-organisation of the system is observed. The system leaves its chaotic state and provides a network of hexagonal running called Bénard’s cells. It the warmth increases, this phenomenon disappears.

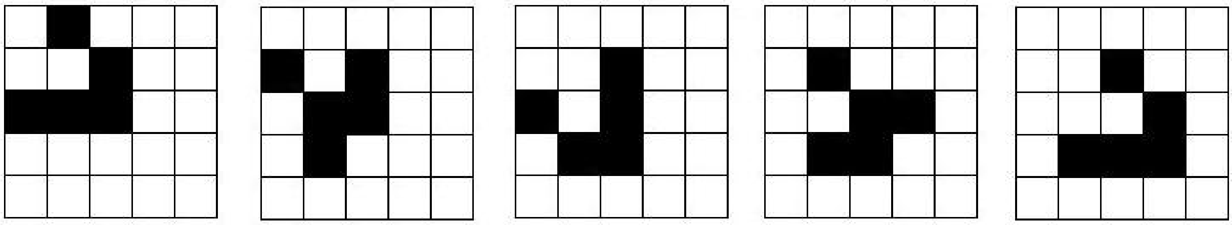


In the game of life from Stanislas Ulam and John von Neumann, a moving pattern becomes apparent to an observer of the grid [31]. The game of life is a grid with cells with a

---

[3] Authors : Di Marzo Serugendo G., Gleizes MP., Glize P., Holvoet T., Karageorgos A., Lopardo G.

connexity of 8. A cell can be alive or dead. The behaviour of a cell follows the three local transition rules:
- if a cell is alive and 2 or 3 of its neighbours are alive it becomes alive else it becomes dead.
- if a cell is dead and 2 or 3 of its neighbours are alive it becomes alive.
- in other cases, the cell does not change its state.
When the system is running, the observer sees a glider.



## 4.3.2. Properties of Emergent Phenomena

This short analysis of the term emergence acknowledges several properties of emergent phenomena.

- Need to be observable at some level
  An emergent phenomenon (a property, a structure, a pattern, a function) is patent for an observer at the macro level of the system.
- Novelty
  The phenomenon -always derived from a particular organization of the micro-level parts- is radically different from their individual properties and cannot be derived or predicted from them. Its identification requires concepts different from the concepts used to qualify the micro-level activities.
- Coherence
  This physical phenomenon can be specifically identified, and consequently a given coherence strongly related -and distinct- from the activities of its parts.
- Irreducibility
  Churchland defines the emergence in terms of irreductibility of the properties of a high-level theory to properties of a lower level theory » [1]. Structures and/or functions appear at a macroscopic level and the observation of the components properties cannot enable to predict them [2]. For example in complex systems, the complexity often results in features, emergent properties, which are properties of the system that the separate parts do not have.
- Interdependency between levels
  If the microlevel is the substrat where the emergent phenomena comes from, the macrolevel phenomena contrain the activities of the part. Thus, there is a strong dependency between the dynamics observed at the micro and micro levels.
- Non linearity
  A chain of linear activities enables explainability and predictability of a collective phenomenon. On the opposite, an emergent one needs non linear activities in the micro-level. Loops of positive and negative feedbacks are typical examples allowing non linearities.

## 4.3.3. System Characteristics

- At least two levels (micro-macro)

If occurring, an emergent phenomenon can be only associated with a particular collective behaviour and thus requires minimally two levels in the system. We could have more than two levels in a system, but emergence is only observed from two consecutive levels.

- Micro-Macro effect
- Dynamical

Because an emergent phenomenon is observable during time, it needs a form of self-maintained equilibrium. Nevertheless it is not a homeostatic but dynamic equilibrium. Emergence occurs in a narrow possibility space lying between conditions that are too ordered and too disordered. This boundary or margin is the edge of chaos [32], which is always far from equilibrium.

Complexity theories are interested in studying this particular equilibrium using new forms of attractors (fixed point, limit cycle, stange attractor) explaining why emergence is not easily predictable. Near these equilibria, a system has the ability to self-organize allowing an emergent phenomenon.

Complexity is the point at which self-organising systems emerge to create new patterns of coherence and structure of relation. Then, complex structures that resist the seemingly inexorable flow of time can be described as emerging self-organising systems. For example, autocatalytic sets are natural models of functional integration. They are functional wholes. Maturana and Varela [35] have formulated the autopoietic systems, in which each part exists both for and by means of the whole, while the whole exists for and by men as of the parts. Autopoietic systems are true sources of emergent properties.

### 4.3.4. Operational Aspects in Computer Science

For ten years new challenges appeared in artificial systems ability to execute themselves, adjustment of their behaviour due to various circumstances, managing in the best their resources and self-repairing when needed. These kinds of requirements are typical of many new concepts, namely: Autonomic Computing, Pervasive Computing, Ubiquitous Computing, Emergent Computation, Ambient Intelligence, Amorphous Computing... These systems are characterized by emergent phenomena derived from: a great number of interacting components (intelligent objects, agents, software components), which number changes at runtime (open systems), the inability to impose a global control, an evolving and unpredictable environment, and a functionally adequate behaviour to achieve in this environment.

Rather than attempting to eliminate such emergence, it could be interesting to explore how this might be deliberately harnessed. That is, address how to engineer artificial systems with desirable emergent properties.

- Decentralised control

There is no entity which controls all the components of the system. Each component is autonomous.

- Interacting parts

There are specific mechanisms to describe the emergent properties of a complex system. The mechanisms dealing with interactions match conveniently into two types: external and internal. The external mechanisms are ways to modify the system from environment (constraints, rules, artifacts). The internal mechanisms are ways to change the interaction dimensions that are unfolded by processes within the system. Interaction is essential to this framework because the events of novelty and innovation within a system arise from the interactions of these agents with each other and with the environment.

- Self-organisation. (see section 4.2)

# 4. Case Studies

## 4.1. Objectives[4]

Researchers in multi-agent systems study more and more self-organisation phenomenon because analogies can be done between the multi-agent-systems and natural systems such as the social animals.
The way to describe systems can be very different and focus on different points of view on the systems: entity design, results, measures… In order to better understand self-organisation, emergent properties and their underlying mechanisms, a framework to describe systems has been proposed. This framework highlights the self-organisation mechanism and the entities which have to self-organise. Our final aim is to complete this framework by a list of measures, which enable to compare different mechanisms.

The framework contains the following parts:

1. **The system description** which presents the purpose and function of the system, as well as its behaviour at global level, the inputs of the system (the data to be given to the system at the initialisation and/or during the functioning)

2. **The environment description** the system is situated in.

3. **The perturbations coming from the environment** which can be a list of the various types of stimuli the system can receive from the environment and how they can possibly vary over time (perturbations)

4. **The entity description.** The designers have to provide a description of the characteristics of the system components (autonomous entities). This should include (but not limited to) the following :
- Goals
- Skills or capabilities (i.e. tasks that the entities can perform in the form of <precondition, action, effect>)
- Beliefs about the others entities
- Beliefs about itself
- Beliefs about the environment of the system

5. **The interaction description** which provides how the system entities interact with one another, i.e. by exchanging communication messages or by causing and observing particular changes in the state of their environment.

6. **The self-organisation engine description** details the engine (mechanism) of self-organisation processing:
- what triggers the re-organisation
- is this a local or a global one?

---

[4] Authors : Di Marzo Serugendo G., Gleizes MP., Karageorgos A.

- is this an internal or external one?
- what are the steps (i.e. interaction sequences) involved in the various self-organisation cases?
- when does the self-organisation process stop?

**7. The characterisation of the self-organisation approach (defined in section 4.2)**

*Absence of external control (autonomy)*

*Decentralised control: not mandatory since we can have an internal centralised control. Not needed in general, but needed in MAS.*

*Dynamic operation (time evolution)*

*Global order endogenous*

*Emergence from local interactions), emergent properties*

*Dissipation (energy usage/far-from-equilibrium)*

*Instability (self-reinforcing choices/nonlinearity), Parameter sensibility*

*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)*

*Simple local rules*

## 4.2. Managing Computers Networks

### 4.2.1. Description [5]

The benchmark considers a set of applications (processes) processing on an open network of workstations, while dynamically satisfying a set of constraints such as:

- dynamic load balancing of processors;
- minimizing communications costs between highly communicating applications
- optimal sharing of resources and data between process belonging to a same application;
- mutual exclusion between applications accessing a same set of resources ;

The openness of the environment implies that during the processing, new workstations (respectively existing workstations) can join (respectively leave) the network. New applications (respectively existing applications) could also be launched (respectively resume their activity). The network could also be subject to perturbations such like workstations breakdowns for example.

Different aspects could be considered in this benchmark, dynamic routing of processors, network topology evolution (specificities of the problem with respect to some kind of

---

[5] Author : Hassas S.

topologies: simplified problem or a more complex problem, etc), prioritizing some constraints satisfaction (different weights accorded to different constraints), etc.

**Data available at the beginning:**
- network topology,
- applications characterization,
- relations between applications,
- relations to resources access

**What the system has to provide:**
The system has to provide the right services.

**The perturbations to take into account**
Breakdowns are generated along stochastic distribution including correlated occurrences of disturbances.

## 4.2.2. Approach Based on Holons Systems [6]

### 5.2.2.1 System Description

*What are the input data ?*
The benchmark considers a set of applications (processes) processing on an open network of workstations, while dynamically satisfying a set of constraints such as:
- dynamic load balancing of processors;
- minimizing communications costs between highly communicating applications;
- optimal sharing of resources and data between process belonging to a same application;
- mutual exclusion between applications accessing a same set of resources.

The data in input are defined by the network topology, node capacities, applications running, ...

*What are the expected results of your system?*
The satisfaction of the set of constraints by the nodes and processes is the result of negotiations in order to assign to each process a node to be executed on. These negotiations modify the overall organization of nodes and processes without external control thus we can say that it's the result of a self-organization process.

*Which techniques do you use?*
We have adopted an holonic perspective to solve this case study and prove the capacity of holarchies to self organization. The notion of holon was originally introduced by Arthur Koestler in 1967 to refer to natural or artificial structures that neither wholes nor parts in an absolute sense [33]. According to Koestler, a holon must respect three conditions:
- being stable : it reacts when strong perturbations are applied,
- having the capability of autonomy : it is able of self-organization to achieve its own goals,
- being capable of cooperation: it works in common projects according to shared goals with otherholons and other layers of holons.

*What are the entities inside the system that have to self-organize?*
We distinguish two main agent types: the process and the nodes.

---

[6] Authors : Gaud N., Rodriguez S., Hilaire V.

### 5.2.2.2 Description

*What is the environment of the system composed of?*
The environment is composed of linked nodes and applications. Each node has a CPU which may execute processes and has resources availables. The links between nodes allows communication and process migration.

### 5.3.2.2 Perturbations Coming from the Environment

*Which kind of perturbations your system can take into account?*
The openess of the environment implies that during the processing, new workstations (respectively existing workstations) can join (respectively leave) the network. New applications (respectively existing applications) could also be launched (respectively resume their activity). The network could also be subject to perturbations such like workstations breakdowns for example.

### 5.3.2.3 Entity Description

*If you use agent please describe them in terms of:*
Each entity which composes a holarchy is called holon. We distinguish a particular kind of holons, which can not be decomposed in sub-holons: atomic holon, classically called agent.

*Process Agent*
A process agent owns a process and handles it in order to optimize the problem constraints. This agent can migrate to another node if it isn't satisfied. It can access information about the ressources needed and the charge necessary to compute its process. It knows also the state of the current node and its neighboors.

*Node Agent*
The node agent is situated on a specific node and communicates with the process agents. It knows the current load of its node and the one of its neighboors.
The node agent is also in charge of managing the access to a resource (mutual exclusion). It also maintains an up-to-date history of modification of its resources.
Before any process reaches its resources, it makes a saving point, to be able to re-generate the data in a coherent state in case an error arises.
Every application may register with node agents managing the resources that it should use. This inscription can be made either at the launch of the process agent, or at the time of a particular request (ex: edition of a file). The node will also to maintain an up-to-date wish-list indicating each process agent susceptible to reach the resources it manages. This list will enable agents to know with whom to group to generate shared accesses to the same resource.

### 5.3.2.4 Interaction Description

Interactions between holons are defined by their roles in the holarchies. The role an holon plays may evolve as shown in figure 1. In the following subsections we sketch the interactions within each holarchy.

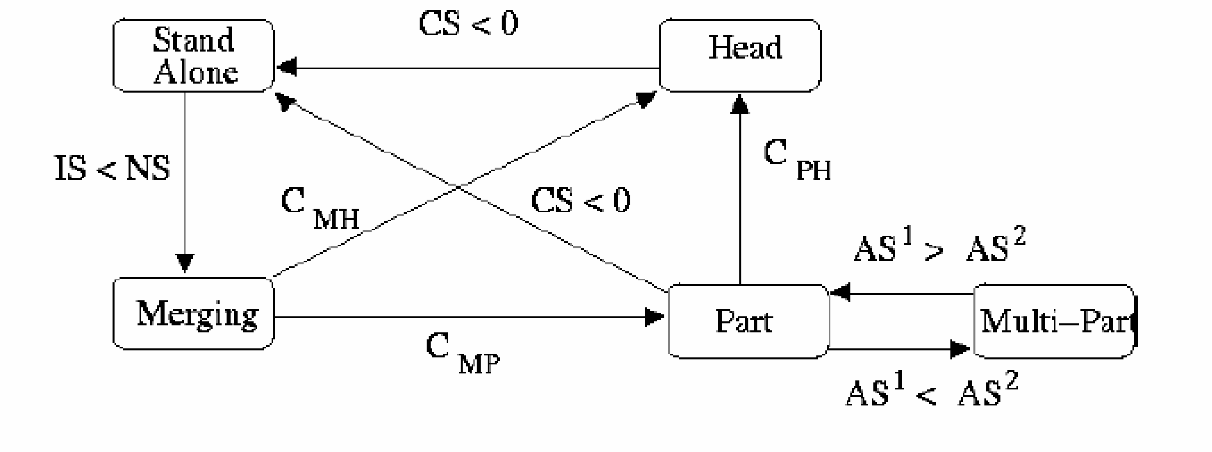


Figure 1: Role transition diagram

*Communication holarchy*
In this holarchy a process agent groups together with those who communicate most with him. The regrouping phase proceeds in two steps:

- They create an holon who will represent them. But they can leave it when they wish.
- As one goes along if they effectively continue to communicate, the link which unites them strengthens.

They will then leave this holon only when the condition which united them, will become false.

*Execution holarchy*
The execution holarchy possesses no regrouping criterion. The holons choose or not to grant to their representative in another holarchy a supplementary role of representative (Head) in this holarchy. They will then be considered by the outside and from the execution point of view as a single process.

*Ressource holarchy*
In this holarchy the holons group together when they wish to read the same resource. They give rise to a representative in charge of reading the resource, then redistributing the read value between the various parts.

The satisfaction engine propose in this work is inspired by our work with holonic Systems [40]. The framework was formalized using the RIO Methodology [29] and we have proved some selforganization properties [41].

*Holon satisfactions*
In order to enable holons to dynamically change their roles, we define a satisfaction based on the progress of his current task. This satisfaction, called instant satisfaction, depends on the played role and is calculated using the following definition.

**Self Satisfaction** ($SS_i$) Satisfaction for the holon i produced by his own work.
Collaborative Satisfaction ($CS^H_i$ ) Satisfaction produced for the holon i by his collaboration with other members of the Holon H. This satisfaction can be either positive, when the other members'work help i in its task, or negative, when the other members' work impose barriers to the achievement of the task.

**Accumulative Satisfaction** ($AS^X_i$ ) Satisfaction produced for the holon i by his collaboration with members of multiple holons. This satisfaction is only used in the Multi-Part Role. When an holon belongs to a holon and is unsatisfied, two options are available. The holon may quit its current holon and look for other group, or it may join a second holon. This satisfaction guides the decision in this situation.

$$AS^X_i = \sum_P CS^P_i \quad where\ X = H_i \tag{1}$$

Instant Satisfaction ($IS_i$) Satisfaction produced by the work done up to the moment

$$\forall i \in HMAS \quad IS_i = \begin{cases} CS_i + SS_i & if\ R_i = Part \vee R_i = Head \\ AS_i + SS_i & if\ R_i = MultiPart \\ SS_i & if\ R_i = Stand - Alone \end{cases} \tag{2}$$

Where Ri is the role played by the holon i.

*Satisfaction of a node agent*

Concerning the node agent, its satisfaction depends on its personal load and the load averages of its neighbours. If its load is superior, its satisfaction increases otherwise it decreases.

Remark : The introduction of the node satisfaction, a priori, accelerate the auto-organization process for the load-balancing.

Definition 6.1 Satisfaction of a node agent Ni

Given Cmoyi the average load of the network perceived by Ni and Cpj the personal load of the node Nj and Neighbourhoodi the set of neighbour nodes of Ni.

$$C_{moy_i} = \frac{1}{n_i} \sum_j Cp_j \text{ with } j \neq i \text{ and } j \in Neighbourhood_i$$

The satisfaction of the node agent $N_i$ is given by :

$S_{N_i} = f(C_{moyi}, Cp_i)$ with f a function to define.

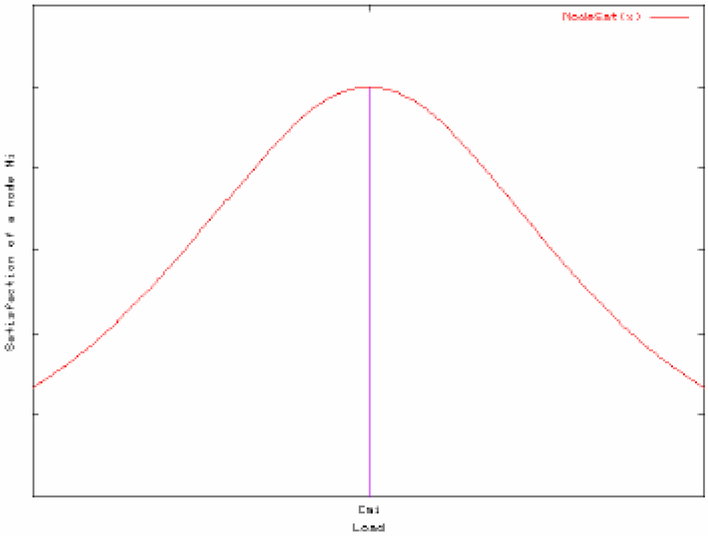


Figure 2: Evolution of the satisfaction of the node Ni

*Satisfaction of a process agent*

The satisfaction of a process agent depends on : in the Ressource holarchy : the time to obtain the desired resource. More he has to wait, more his satisfaction falls.
in the Execution holarchy : the quotient between the allocated processor's time and that which it wishes and the satisfaction of the node where it is executed. in the Communication holarchy: the cost of the various communications which he owes realized.
Knowing that the communication cost depends on the time of packet routing.

Definition 6.2 Satisfaction of a process agent pi

$$S_{p_i} = f(S^L_{p_i}, S^C_{p_i}, S^R_{p_i}), \text{ with f a function to define.}$$

$$\text{and with} \begin{cases} S^L_{P_i} & = f_1(S_N, MR_{available}, MR_{wished}) & \text{Load-balancing satisfaction} \\ S^C_{P_i} & = f_2(Cost_{Com}, K) & \text{Communication satisfaction} \\ S^R_{P_i} & = f_3(t_{rsc_i}, \forall rsc_i \in RSC_i) & \text{Resource satisfaction} \end{cases}$$

Given

$S_N$ : the satisfaction of the node where $p_i$ is running.

$MR_{available}$ : the volume of material resources (cpu, memory) attributed to $p_i$.

$MR_{wished}$ : the volume of material resources (cpu, memory) wished by $p_i$.

$Cost_{Com}$ : the communication cost.

$K$ : Influence of the parent holon (cf. ?? page ??).

$t_{rsc_i}$: the time to obtain the software resource $rsc_i$, with $RSC_i$ the set of all needed resources.

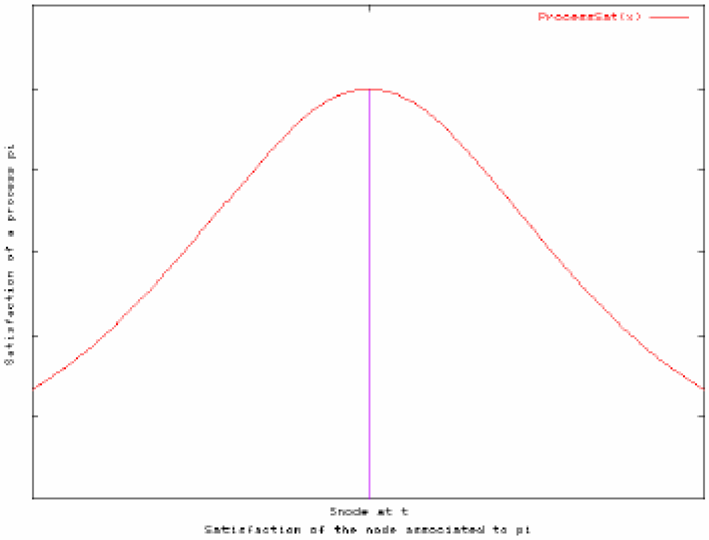


Figure 3: Evolution of the process satisfaction according to the satisfaction of the node Ni

*Agents affinities*
Process Agent (boolean) affinity defined by

- execution order/preferences
- resources
- communication threshold

Node Agent defined by immediate neighborhood

*An example of the self-organisation mechanism*

This example describes the system’s functioning to integrate a new node Nk to the network. Given Nl, one of the neighbour of the new node Nk. The figure 4 presents the evolution of the satisfaction of the node Nl according to the average network load that it perceives. The figure 5 presents the evolution of the satistaction the process pi migrating from Nl to Nk.

1. At t = t0: the system is in a stable state.
2. At t = t1: a new node Nk join the network, it informs its neighbours that it has just joined the network.
3. The satisfaction of its neighbours falls and those of their process agents also.
4. At t = t2: some process agents have begun to migrate to Nk.
5. The satisfaction of Nk decreases as processes arrive, the process agents thus stop migrating to Nk.
6. At t = t3: The system reaches a balance, the migration is ended and the load is stabilized.

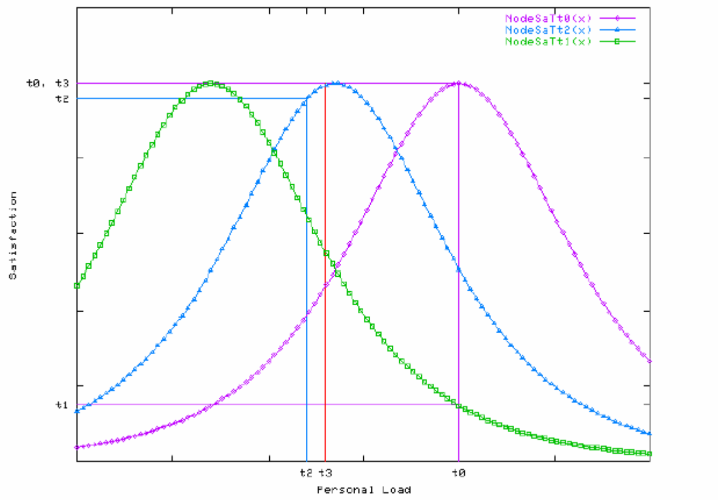


Figure 4: Evolution of the satisfaction of the node Nl

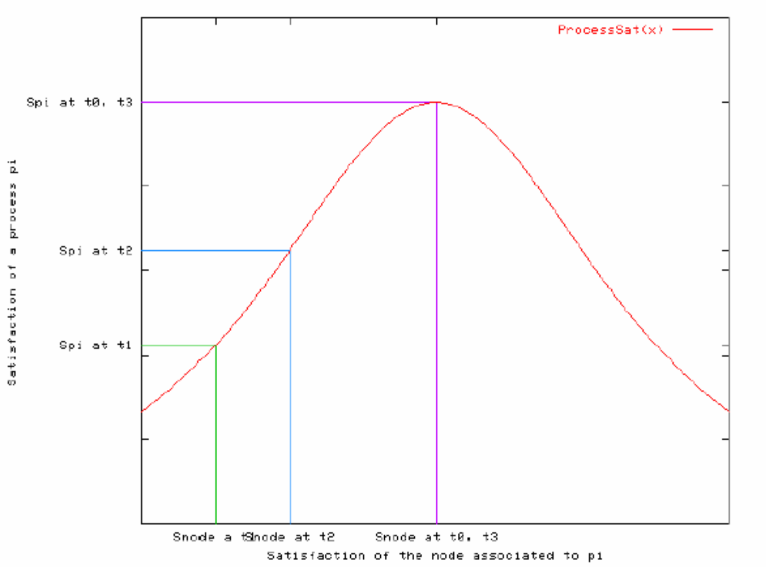


Figure 5: Evolution of the satisfaction of the process pi

## 4.2.3. Approach Based on Peer to Peer Systems [7]

### 5.3.2.5 System Description

Peer–to–peer (P2P) systems comprise nodes connected in a network and functioning in a decentralized manner with the aim to increase the potential of the whole system to serve its "customers" more efficiently and effectively. Each node has the potential to play any role from a set of roles (e.g. being a server, a client, a gateway or non-gateway), although the physical characteristics and processing capabilities of nodes may differ. These systems reside on the edge of ad-hoc networks.

In such a setting we study the self-organization of peer-to-peer systems at various levels: at the communication, task and roles' layers (i.e. concerning the roles activated/deactivated in the system and their dependencies).

At the communication layer the system organizes itself by computing the set of gateways through which all nodes communicate. At the task layer nodes exploit the communication infrastructure to establish groups of nodes that work jointly to achieve a goal or perform a task. At the roles layer agents are assigned roles either to achieve communication goals, or to perform tasks. In the latter case, roles are application/domain dependent.

Each peer is considered to be an agent that may work in conjunction with other peers to serve systems' "clients".

### 5.3.2.6 Environment Description

Each node has a number of neighbours. The number of neighbours of a node may differ from time to time. We may consider that the whole system is "surrounded" by clients that may request "goods" (e.g. data, information items, meta-data, computing resources or just the routing of messages).

### 5.3.2.7 Perturbations Coming from the Environment

- A new node joints the system
- A node leaves the system
- A node receives a request
- A new communication link is established/destroyed

### 5.3.2.8 Entity Description

The functions of each node are realized by an agent.

Agents, being minimal are able to
- communicate with neighbours
- set their communication status (gateway or no-gateway)
- form/propagate response/request messages
- lock resources to clients

Agents possess knowledge about
- the status of their neighbours (e.g. roles that they play)
- routing of requests for locating "goods"
- availability of its resources
- availability of others' resources (e.g. using an aggregation method that allows it to apply a searching method – no node has complete knowledge of the system)

---

[7] Author : Vouros G.

Furthermore, beyond minimal agents, we may consider agents that have more advanced abilities concerning:
- means-end reasoning (planning for achieving goal states)
- group decision making
- deliberation (individual and social) (assessing options and deciding which one to pursue)
- collaboration with peers towards achieving a goal or performing a task.

#### 5.3.2.9 Interaction Description

Agent can send messages to other agents

#### 5.3.2.10 Self-Organisation Engine

At the communication layer, agents gather information of their 1 or 2 –hops away nodes to determine their gateway status. This is done in a totally distributed way and dynamically: the nodes may change their status (role) according to the perturbations from the environment.

At the task layer, agents that receive clients' requests try to locate agents that match capabilities requirements towards serving the request. Agents that fulfil capabilities requirements are determined to play specific task-dependent roles. This is a dynamic process: Due to perturbations from the environment the system may re-organize itself by employing new roles that a new set of agents may play.

### 4.2.4. Approach Based on Routing and Delegation Concept [8]

#### 5.3.2.11 System Description

The solution is freely inspired by routing techniques. We expect good properties of these techniques can be inherited here but such conjectures have to be pragmatically verified by running what we propose on a shared testbed. Our proposition here is only a first draft, this is not based on any bibliography about routing techniques. This is not implemented and must not be considered as a finalized work.

We associate an "agent" to every processor present in the network and the key feature we try to design is a way to route new computational needs. By routing, we mean that each agent may commit to a task and/or delegate it, i.e. "route" the task to himself or neighbours. Therefore, he needs to maintain preferential delegation targets within its direct neighbours. We believe our proposal might lead to a self-organisation result as it consists in a constant adaptation of delegation trails, and we consider delegation trails is an organisation between agents.

By considering a new computational need is inserted at a specific point and its result must arrive at some recipients, we think the problem is pretty close to the routing issue with the following constraints added :
- the task must be computed on the way,
- some special resources must be on the way.

We describe fixed agents with a strategy to keep memories of good and bad direct delegation experiences. This memory determines the delegation activity through the network and then constitutes the organisation. We here described a simplified scheme where only atomic computational needs are inserted.

#### 5.3.2.12 Environment Description

---

[8] Author : Deguet J.

Each agent perceives only a part of the environment:
- messages from others, via the network interface,
- the local processor's schedule which contains atomic tasks.

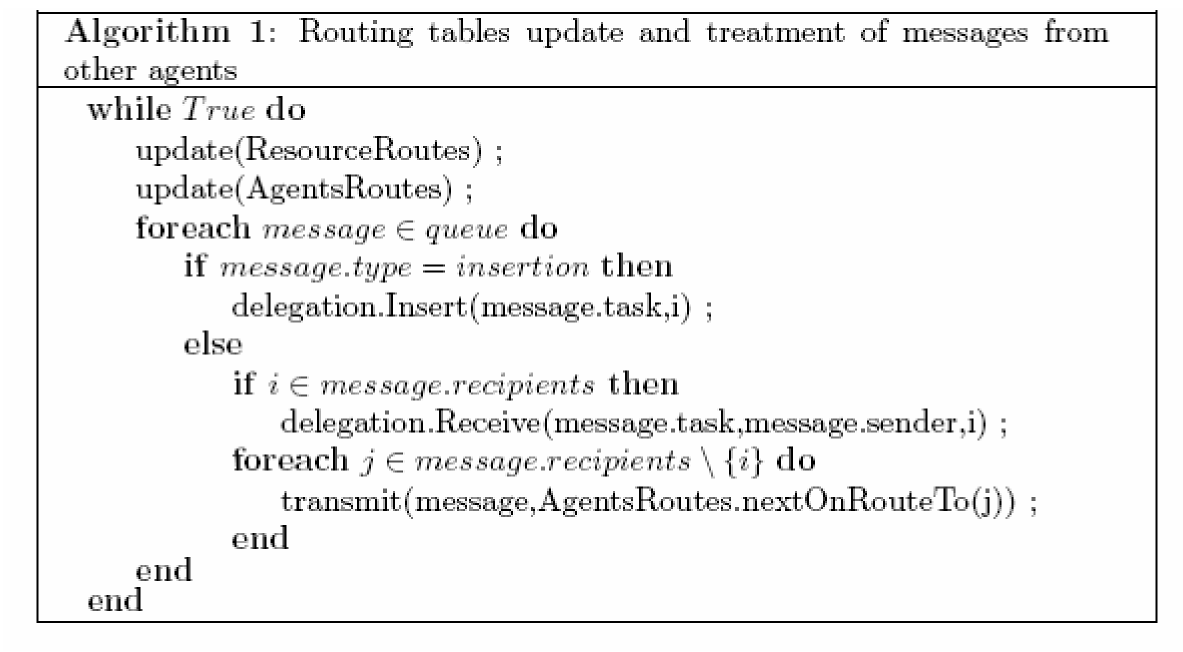
Each processor executes the loop:
1. unqueue next task with needed resources available,
2. execute it,
3. ask the agent to send result to the recipients list, i.e. where the result is needed in the network

#### 5.3.2.13 Perturbations Coming from the Environment

Every kind of disturbance is supposed possible such as:
- addition/removal of needs
- addition/removal of computers
- communication cut
- resource vanishment

#### 5.3.2.14 Entity Description

The agent here simply maintains a delegation willingness table which can be interpreted as its belief about delegation benefit.
At the message level, special treatment is applied to incoming results and needs, in addition to maintaining routing tables to agents and resources.



**Algorithm 1:** Routing tables update and treatment of messages from other agents

```
while $True$ do
    update(ResourceRoutes) ;
    update(AgentsRoutes) ;
    foreach $message \in queue$ do
        if $message.type = insertion$ then
            delegation.Insert(message.task,i) ;
        else
            if $i \in message.recipients$ then
                delegation.Receive(message.task,message.sender,i) ;
            foreach $j \in message.recipients \setminus \{i\}$ do
                transmit(message,AgentsRoutes.nextOnRouteTo(j)) ;
            end
    end
end
```

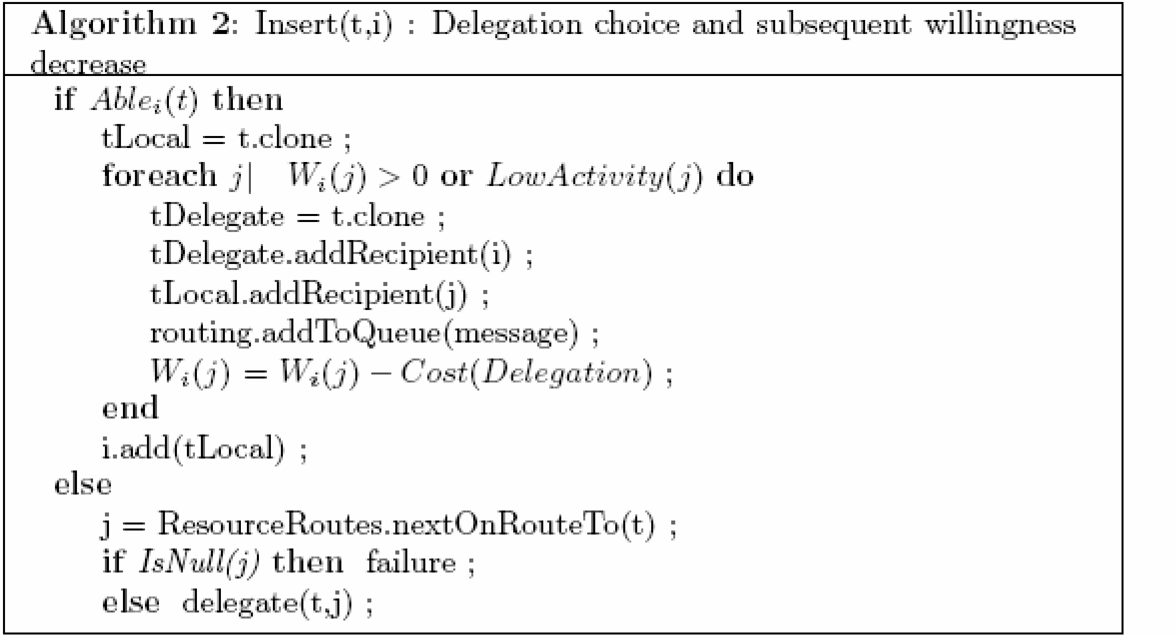
#### 5.3.2.15 Interaction Description

We consider a message between agents is a construct:
- sender contains the sender identifier
- type contains either :
- insertion if the message contains a task to execute
- result if the message contains the result of the task
- task contains the task description

- result contains the result if computed
- recipients contains the list of all recipients (identifiers) for the result

#### 5.3.2.16 Self-Organisation Engine

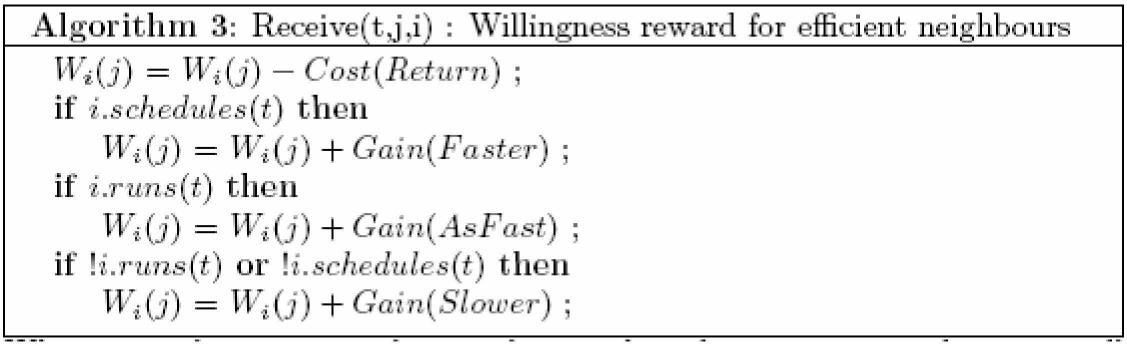
The “engine” is constituted of two steps, one achieved at delegation time and the other when a result is perceived :



**Algorithm 2**: Insert(t,i) : Delegation choice and subsequent willingness decrease

```
if $Able_i(t)$ then
    tLocal = t.clone ;
    foreach $j \mid W_i(j) > 0$ or $LowActivity(j)$ do
        tDelegate = t.clone ;
        tDelegate.addRecipient(i) ;
        tLocal.addRecipient(j) ;
        routing.addToQueue(message) ;
        $W_i(j) = W_i(j) - Cost(Delegation)$ ;
    end
    i.add(tLocal) ;
else
    j = ResourceRoutes.nextOnRouteTo(t) ;
    if $IsNull(j)$ then failure ;
    else delegate(t,j) ;
```

The Able function is true when the agent i can execute task t i.e. no special resource is needed. If not, it directly delegates to the next agent on the route to the needed resources. Else, the agent does commit to the execution of the task as well as he delegates to “good neighbours” (and substract to the willingness Wi(j) as we a priori increased the load by delegating), resulting in some competition. The second part determines who won the game of computing the task the faster:



**Algorithm 3**: Receive(t,j,i) : Willingness reward for efficient neighbours

```
$W_i(j) = W_i(j) - Cost(Return)$ ;
if $i.schedules(t)$ then
    $W_i(j) = W_i(j) + Gain(Faster)$ ;
if $i.runs(t)$ then
    $W_i(j) = W_i(j) + Gain(AsFast)$ ;
if $!i.runs(t)$ or $!i.schedules(t)$ then
    $W_i(j) = W_i(j) + Gain(Slower)$ ;
```

When a result is perceived, we rely on what the processor is doing or still schedules to determine if we have been right to delegate.

*Some conjectures*

- distant computers delegate to each other via propagation of positive willingness (delegation trails)
- an unreliable computer i makes Wx(i) decrease but have high Wi(x)
- an unreliable link between i and j makes both Wi(j) and Wj(i) decrease

- if Wi(j) is positive, Wj(i) is negative (no perpetual delegation)
- with the cost of delegation, we avoid to delegate to dead computers for too long
- W is a good charge estimator so the LowActivity flag is almost never used

## 4.2.5. Approach Based on Stigmergy Mechanism [9]

### 5.3.2.1 System Description

*What are the input data ?*

Criteria to satisfy

- load distribution,
- membership of processes (same application, strongly communicating applications, competitive applications, etc)

*What are the expected results of your system?*

Emergence of spatial aggregate of processes on a set of processors, satisfying the specified criteria (a spatial organization)

*How this result can be viewed as a self-organisation result:*

The result is obtained just by applying local rules for behaviours, without any external control. It is expressed as a situation where the systems dynamics guide it to reach an attractor where the specified criteria are balanced. The attractor corresponds to the spatial organisation of the processes in such a way that the multiple criteria are satisfied.

*Which techniques do you use?*

We use Stigmergy. Our approach is an ant-like based approach (foraging behaviour). We use multiple electronic pheromones to represent the different criteria to be satisfied. Ant-like processes agents move over the network following the gradient of their specified peromone.[21]

*What are the entities inside the system which have to self-organize?*

Agents associated to processes

### 5.3.2.2 Environment Description

*What is the environment of the system composed of?*

Environment is a dynamic network of processors (a dynamic graph)

### 5.3.2.3 Perturbations Coming from the Environment

*Which kind of perturbations your system can take into account?*

Processors breakdowns, sudden overloading of the network, arrival of new nodes (processors), departure of existing nodes (processors),..etc.

### 5.3.2.4 Entity Description

*If you use agent please describe them in terms of:*

- *Individual goal :* no individual goal
- *Skills or capabilities (i.e. tasks that the agent can perform in the form of*
    - <if perception of an appropriate pheromone field, follow the gradient field with a high probability, reinforce the pheromone amount> :

---

[9] Author : Hassas S.

    - <if no perception of an appropriate pheromone field, move randomly, no effect>
- Beliefs about the other agents or accointance : no
- Beliefs about itself : no
- Beliefs about the environment of the system (so without the others agents) : no

#### 5.3.2.5 Interaction Description

*Which kind of communication?*
Interactions mediated by the environment (pheromone field)

#### 5.3.2.6 Self-Organisation Engine

Engine: The self-catalytic mechanism of ant-like foraging behaviour

- what elements trigges the re-organisation : perturbations of the environment
- is this a local or a global one? Global to the system
- is this an internal or external one? Internal reorganization
- What are the steps (i.e. interaction sequences) involved in the various self-organisation cases? Pheromone diffusion, evaporation, enrolling mechanism (self-catlytic behaviour)
- When does the self-organisation process stop? When the system is completely stable, and no perturbation occurs

## 4.2.6. Characterisation of the Self-Organisation Approach

| **Characterisation criteria** | Holon approach | Peer to peer systems | Routing and delegation concept | Stigmergy |
|---|---|---|---|---|
| *Is an external control (autonomy) absent ?* | Yes | Yes. | Yes, because we consider that the only external input is the need which is hardly interpreted as control | Yes |
| *Is there a decentralised control?* | Yes | *Yes. (in case a client request is served by more than one agents, then there is one agent – peer – that plays the role of "coordinator". However, this is dynamically determined).* | Yes, as decisions are taken locally and no agent has a special behavior | Yes |
| *Dynamic operation (time evolution)* | Yes | Yes. | Yes, we hope that kind of solution to exhibit a | Yes |

| | | | | |
|---|---|---|---|---|
| | | | constant adaptation through organisation | |
| *Endogenous global order* | ?? | Yes | ?? | Yes |
| *Emergence from local interactions, emergent properties* | To be tested | *Yes, (there is no node that has complete knowledge of the system, tasks performed and structures)* | ?? | Yes |
| *Dissipation (energy usage/far-from-equilibrium)* | *No* | *Yes (new requests, nodes and communication links make the system to behave in a dynamic way).* | ?? | Yes |
| *Instability (self-reinforcing choices/nonlinearity), Parameter sensibility* | *Affinity/satisfactions dependent* | No. | ?? | Yes |
| *Multiple equilibriums (many possible attractors)* | To be tested | *YES (each node may play more than one roles concurrently)* | ?? | *Yes (as it could happen in ant foraging behaviour)* |
| *Criticality (threshold effects/phase changes)* | To be tested | ?? | ?? | *Yes (as it could happen in ant foraging behaviour)* |
| *Redundancy (insensitivity to damage)* | *Seems to* | Yes. (through continuous update of nodes' status) | No | Yes |
| *Self-maintenance* | Seems to | Yes | No | Yes |
| *Adaptation (functionality/tracking* | Yes | Yes | ?? | Yes |

| | | | | |
|---|---|---|---|---|
| *of external variations)* | | | | |
| *Complexity (multiple concurrent values or objectives)* | Yes | Yes.If optimization requirements are set. For the system described there are no optimization requirements | No | Yes |
| *Hierarchies (multiple nested self-organised levels)* | Yes | Yes | No | No |
| *Simple local rules* | Yes | *Yes. (mostly at the communication layer)* | Yes | Yes |

## 4.3. Manufacturing Control

### 4.3.1. Description [10]

The benchmark comprises the coordination and control of the internal logistics of a manufacturing department. Finished parts have to be delivered against a given due date for assembly in another department. Parts are transported in containers. Typically, a container holds about 10 parts, but this may vary. The system comprises a grid of container storage spaces, distributed across the manufacturing department. The system also comprises 10 workstations with varying properties and capabilities. Workstations have two or three locations at which a container can be placed. An operator picks parts out of one container, processes them in a pipelined fashion on the machines in his/her workstation, and places processed parts in another container at her/his workstation. An automated transporter, moving over rails, normally transports the containers. It can carry maximally two containers at any given time. The parts enter the system in a container through the storage system. The system produces a mix of products, imposing varying loads on the workstations. The transport system has sufficient capacity on average but intermittently experiences rush hour complications. Operators have varying qualifications (having received training in function of shortages that occurred historically). The operator schedule is given.

**Simplified hypothesis**

All the parts in a container are the same.

A container must be manipulated by several given workstations according to the process the parts have to follow. This process consists in a following of jobs to be done and it can be known by the container.

In a workstation, we need to have one empty container to put the treated part. A workstation consists in N given machines able to perform N tasks.

An operator is affected to a workstation in function of his capabilities and had constraints such has his breaks, his holidays, his work duration. He can be ill and not available.

**Data available at the beginning**

---

[10] Author : Valckenaers P.

The system has to start from an initial description of the factory:

- number of workstations,
- description of the workstations capabilities,
- number of operator,
- description of these operators in terms of capabilities and constraints,
- number of containers
- the process associated to each container,
- data files on order arrival of containers

**What the system has to provide:**
The system has to find the right organization of these elements in the factory to have an efficient functioning.

**The perturbations to take into account**
Machine breakdown, process failures, operator unavailability, operator switch (other skills), process time variations, order arrival, rush order arrival, etc.
Breakdowns, failures…are generated along stochastic distribution including correlated occurrences of disturbances.

## 4.3.2. Approach Based on Ants Algorithms [11]

### 5.3.2.1 System Description

*What are the input data ?*

- Manufacturing system definition
    - Workstations (type, number, location, capabilities…)
    - Storage system (type, number of spaces, location of spaces…)
    - Transport system (type, connections, capabilities…)
    - Operators skills and availability

- Product type definitions
    - Recipes

- Order book definition
    - Arrival time and product type

- Manufacturing scenario
    - Break down statistics
    - Process failure statistics
    - Process time variation statistics

*What are the expected results of your system?*

- Effective supervision of the manufacturing activities
- Availability of reliable short-term prediction of the state of the production system, usable to plan secondary activities
- Minimization of the required workforce to perform the work on time.

*How this result can be viewed as a self-organisation result?*

- None of the agents can have a global view or data model

---

[11] Author : Valckenaers P.

- No centralized supervisor is allowed
- It is essential not to expose individual software entities (agents) beyond their own manageable and stable scope. For instance, an agent managing a workstation is reusable wherever such a workstation is present in whatever manufacturing system. Software maintenance is only required when the workstation itself changes and the maintenance effort should be proportional to the chance. Yet, the interactions amongst such agents yield an effective manufacturing control that anticipates the immediate future as a means to coordinate the activities.

*Which techniques do you use?*
- stigmergy,
- holonic system >> a specific agent society structure
- delegation whenever something is outside an agents' scope
- low and late commitment (but early preparation when feasible/doable)

*What are the entities inside the system that have to self-organize?*
- Orders amongst each other
- Orders with resources.

#### 5.2.2.3 Environment Description
*What is the environment of the system composed of?*
- Currently implemented as a MAS, following PROSA architecture
- Analysis shows that resource agents can be split in an environment entity (choice-free) and an accompanying (pure) resource agent (handling choices and decisions concerning the resource).
- Implementation environment is a software toolkit/library on top of Java/Eclipse.

#### 5.3.2.2 Perturbations Coming from the Environment
*Which kind of perturbations your system can take into account?t*
- Machine breakdowns
- Process failures >> possible rework and/or rush orders
- Test outcomes >> possible rework and/or rush orders
- Late deliveries
- Order cancellations
- Rush orders
- Variable process duration
- System reconfigurations (new equipment, equipment removal…)
- Degrading equipment and inverse
- Operator (un)availability
- …

#### 5.3.2.3 Entity Description
*If you use agent please describe them in terms of:*
Agents are described in 10]

#### 5.3.2.4 Interaction Description
*Which kind of communication do agents use?*

- Peer-to-peer message between agents

- Stigmergy

#### 5.3.2.5 Self-Organisation Engine

*Describe the engine (mechanism) of self-organisation processing*

There is a self-reinforcing process that builds load forecasts for resources and route/processing forecasts for orders.

i. Orders inform the resources about their intentions (make the necessary bookings) allowing these resources to predict their near-future loads.
ii. Resources inform 'ant agents' acting on behalf of the orders about expected availability of processing capacity, duration, result, etc. allowing these orders to optimise and select intentions.

When the orders inform the resources, the forecast accuracy of the resource improves, allowing the resource to give more accurate answer to queries by the orders, which improves the forecast accuracy of the order, informing the resources more accurately about its intentions, thus closing a loop that improves accuracy until uncertainty, noise and disturbances start to impose an upper bound.

#### 5.3.2.6 Characterisation of Self-Organisation Approach

*Discuss if the following properties (characterisation criteria) can be found or not in the system answer by yes or No and add eventually explanations*

*is the external control (autonomy) absent? Yes*

*is the control decentralised control? Yes*

*Dynamic operation (time evolution) Yes*

*Endogenous global order >> Forecasts Yes.*

*Emergence from local interactions, emergent properties Yes*

*Dissipation (energy usage/far-from-equilibrium) Yes (more complex than well-known biological examples).*

*Instability (self-reinforcing choices/nonlinearity), Parameter sensibility*

*Social control is needed/provided to handle stability issues.*

*Multiple equilibria (many possible attractors) Definitely.*

*Criticality (threshold effects/phase changes) Under investigation.*

*Redundancy (insensitivity to damage) Yes but not to agents themselves (members of a single agent 'colony', not a fully open society).*

*Self-maintenance (repair/reproduction metabolisms) N/A*

*Adaptation (functionality/tracking of external variations) Yes*

*Complexity (multiple concurrent values or objectives) Yes*

*Hierarchies (multiple nested self-organised levels) Not in this case study.*

*Simple local rules. Deep local knowledge (within own scope) but simple/robust toward the remainder of the world.*

## 4.3.3. Approach Based on Cooperative Agents (AMAS Theory)[12]

### 5.3.3.1 System Description

*What are the input data ?*

Input data that can be found in the system description are the following:

- the workstations' properties and capabilities,
- the operators' qualification and schedule,
- the containers (capacity, location, contents); we suppose that all containers have heterogeneous parts,
- the parts (characteristics, process, state) ; we do not consider them separately from the container,
- the automated transporter (location, contents, properties, …) ; we do not consider it in the first approach of the problem,
- some global constraints ; we do not take them into account in a first approach,
- human-handled carts (location, ….) ; we do not take them into account in this first approach.

*What are the expected results of your system?*

The aim of the system is to maximize the outputs (the number of containers processed) of the factory while minimizing the production time. The global solution must respect the time constraints given to all containers.

*How this result can be viewed as a self-organisation result?*

During the process, there are many perturbations coming from unexpected events (machine breakdown, process failures, operator unavailability). Faced with such perturbations, the workstations and the containers have to reorganize their activities without an external centralized control.

*Which techniques do you use?*

We use cooperative agents (according to the AMAS theory) [15], [27] to solve this problem.

In this approach, we want to build artificial systems having a coherent collective behaviour [5] (such systems are said "functionally adequate") whereas agents only seek to reach an individual objective. To obtain such systems every agent is autonomous, has a local view of its environment, and follows a classical lifecycle (perceive, decide, act) while keeping cooperative relations both with each other and with its environment. Definition of cooperation is not a classical one; it relies on three conditions: (1) all perceived signals must be understood, (2) the information provided must be useful for the agent's reasoning and (3) this reasoning leads to useful actions towards other agents.

So, when an agent perceives a cooperative situation it executes the function for witch it has been designed and when the agent perceives a non-cooperative situation, it acts in the world in order to come back to a cooperative state by a self-organization process.

---

[12] Authors : Camps V., Capera D., Gleizes MP.

*What are the entities inside the system which have to self-organize?*
Containers with parts they contain (the number of parts in a container can be null) and operators have to self-organize in order to get an organisation allowing an efficient (optimized) supply chain.
In our approach, workstations are only considered as resources in the system; they do not have a goal to pursue and are not autonomous.

#### 5.3.3.2 Environment Description

*What is the environment of the system composed of? (list the elements)*
The environment of the system is only composed of the factory manager.

#### 5.3.3.3 Perturbations Coming from the Environment

*Which kind of perturbations your system can take into account?*
In this case study, several perturbations can be taken into account, such as process failures, process time variation, order arrival, rush order arrival, unavailability of a workstation (operator illness, strikes, machine breakdown…), etc.

#### 5.3.3.4 Entity Description

*If you use agents please describe them in terms of:*
We have two different types of agents in this case study: container agents and operator agents.

- Their individual goals

Containers: each container has to find a right operator on a right workstation according to constraints (time,…)
Operators: each operator has to find a right container on a right workstation according to constraints

- Their perceptions

Each agent has a local view of the system according to its location.

- Their skills or capabilities (i.e. tasks that an agent can perform in the form of <precondition, action, effect>

Containers: a container agent is able

*to determine which type of workstations can process the parts it contains*

*to find workstations which have a right operator*

*to book a workstation*

*to have a partnership with an operator agent in a given workstation*

Operators: an operator has its abilities

- Their beliefs about other agents or acquaintances

At the beginning agents (containers or operators) do not have knowledge about others. When they encounter other agents, they learn about them (with a limited size memory).

- Their beliefs about themselves

Containers: a container agent knows the process needed to treat the parts it contains
Operator: an operator agent knows its capabilities and its availability

- Their beliefs about the environment of the system (so without the other agents)

They do not have any knowledge about the environment of the system.

#### 5.3.3.5 Interaction Description

*Which kind of communication do agents use?*
In a workstation, agents can communicate in a direct way, by message sending. They also can communicate in an indirect way (by putting booking intentions in the environment).

### 5.3.3.6 Self-organisation Engine

*Describe the engine (mechanism) of self-organisation processing.*

The engine of self-organisation process is the cooperation (see AMAS theory). Every agent tries to find a workstation and to have a relevant partnership (container-operator) in the organisation.

*What elements trigger the re-organisation?*

The fact that an agent (operator or container) is not satisfied because its constraints are not satisfied. It tries then to find a partnership.

*Is this a local or a global one?*

This is a local criterion.

*Is this an internal or external one?*

This is internal.

*What are the steps (i.e. interaction sequences) involved in the various self-organisation cases?*

Every agent (operator or container) has to find a suitable workstation and partnership that satisfy its constraints.

*When does the self-organisation process stop?*

The self-organisation process stops when every agent has found a partnership in a particular workstation with a minimum constraint relaxation.

## 4.3.4. Characterisation of the Self-Organisation Approach

| **Characterisation criteria** | Holon and stigmergy approach | Cooperative Agents Approach |
|---|---|---|
| *Is an external control (autonomy) absent ?* | Yes | Yes. All data inputs are given and there is no interaction with the environment of the system. Every agent is an autonomous entity that takes its own decisions. |
| *Is there a decentralised control?* | Yes | Yes. Every agent has a limited knowledge about the global problem. |
| *Dynamic operation (time evolution)* | Yes | Yes. The system gives solutions that permanently evolve. Solutions are built in a dynamic way. |
| *Endogenous global order* | Yes | Yes. The organisation obtained is the result of the self-organising process. |
| *Emergence from local interactions, emergent properties* | Yes | The right organisation emerges from the reorganisation of the agents (operators, containers) that try to have a better partnership when they are not satisfied by the current one. |
| *Dissipation (energy usage/far-* | *Yes (more complex than* | Yes. The equilibrium can be lost |

| | | |
|---|---|---|
| *from-equilibrium)* | *well-known biological examples).* | by a perturbation; an agent can be then faced with a situation that is non cooperative from its viewpoint. |
| *Instability (self-reinforcing choices/nonlinearity), Parameter sensibility* | *Social control is needed/provided to handle stability issues.* | Unknown a priori. |
| *Multiple equilibriums (many possible attractors)* | Definitely | Yes, depending of the problem |
| *Criticality (threshold effects/phase changes)* | Under investigation | Unknown a priori. |
| *Redundancy (insensitivity to damage)* | *Yes but not to agents themselves (members of a single agent 'colony', not a fully open society).* | Yes. If there are more workstations than necessary, supplementary workstations can be used if a machine breakdown occurs. |
| *Self-maintenance* | N/A | No. |
| *Adaptation (functionality/tracking of external variations)* | Yes | Yes. Agents must adapt themselves to unpredictable events (perturbations). |
| *Complexity (multiple concurrent values or objectives)* | Yes | Yes. The system is complex and non-linear. |
| *Hierarchies (multiple nested self-organised levels)* | Not in this case study | No. |
| *Simple local rules* | Deep local knowledge (within own scope) but simple/robust toward the remainder of the world | Yes. Every agent, whatever its type is, tries to be cooperative with its environment and uses simple local rules. |

## 4.4. Space Conformation of Molecules

### 4.4.1. Description [13]

**Search for Drugs**

Structural Biology is interested in the relationship between the structure of molecules and their biological function. In a general way, questions more often depend on the field of pharmacology. Thus for a known membrane receiver, finding a ligand, i.e. a complementary medicamentous molecule is problematic. It is the docking problem, i.e. the whole set of mechanisms and interactions that play a part during molecular complexes' formation.

Simulation study of the docking, rather than experimental one, has a long history and most algorithms are daily used in the academic environment as well as in pharmaceutical laboratories. Molecular docking is the centre of practical applications like proteins

---

[13] Author : Glize P.

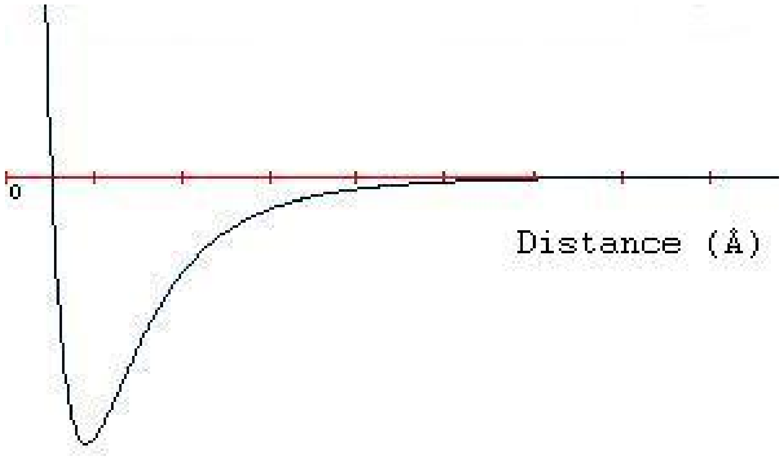
engineering, drugs design and screening of molecules potentially relevant. It has thus already contributed to new ligands design for anti-Aids and anti-cancer agents, and for the treatment of the diabetes. To obtain significant progress, we need to develop new methods to facilitate the techniques of docking by improving our capacities to understand and analyze docking interactions and to develop assumptions for which molecules could have better interactions. Knowledge of the process of space conformation of interacting molecules (folding) is essential and is the subject of this case study.

### Atomic Interactions

A molecule is an assembly of atoms by covalence connections. In fact, strong connections define quasi-stable inter-atomic distances (in the region of the Angstroms) for a particular pair of atoms. Such strong connections are supplemented by strict directional impositions due to weak connections. These weak connections, to have an unspecified effectiveness, can act only at short distances, and as great number.

A molecule's space conformation results from the inter-atomic distances defined by the strong connections and from interactions due to weak connections. To simplify, a space conformation consists in minimizing the residual weak energy on the whole molecule, while respecting constraints of strong connections. Emergence theories will have to notify this objective for this case study.

In inter-atomic relations, the fundamental role is held by outer-shell electrons, pertaining to orbital most external of interested atoms. A union between atoms results from modifications touching with the distribution in space of their outer-shell electrons. A covalence connection is a connection due to a bilateral pooling of electrons. When two atoms approach one another, an attraction between the nucleus of the one and the electrons of the other (and vice versa) appears starting from a certain distance (1 Nano meter approximately). This attraction sensibly increases when distance separating nuclei is about 500 Pico meters. But, starting from this distance, electronic obstruction and repulsion forces between charges of the same sign tend to compensate attraction. A balance is established at a defined distance, characteristic of each atom: it is called Van der Waals' ray. This value is, for example: 120 pm for hydrogen, 140 pm for oxygen, 285 pm for sulphur. Van der Waals' forces induce a weak energy (about 1kcal/mol). Thus they are interesting for molecular structures cohesion only if they are numerous and applied between neighboring atoms. The energy function for weak connections follows a law as indicated on the following curve. X-coordinate represents inter-atomic distance (in Angstroms).



### Resolution of Space Conformation

Molecule's space conformation research (to find the minimum of residual energy) is an NP-Complete problem i.e. the solution cannot be found in a polynomial time according to the number of atoms in the molecule. Resolution of this type of problem must satisfy the following constraints:

- Calculation must be distributed (physically or logically) in such a way that the local algorithm does not have all the information about the molecule.
- Calculation must be emergent i.e. local calculations are unaware about information on the total residual energy of the molecule.
- Ideally (a sensational result!!) you could show (by experimentation or formally) that your algorithm finds a solution in a time limited by a polynomial (depending on the number of atoms).

The system has to start from an initial structure: a space conformation and

**Data available at the beginning:**

- The distance between the atoms with strong connections
- The energy functions between two atoms.

**What the system has to provide:**

The system has to find the right space conformation.

**The perturbations to take into account**

In a first step, there is no need for perturbations. When the problem will be solved, the system can e perturbed in moving the location of one or several atoms in the molecule.

### 4.4.2. Approach Based on Cooperative Agents (AMAS Theory) [14]

**System Description**

*What are the input data ?*

The input data is a description of a molecule in the PDB (Protein Data Bank) format. A PDB file includes as many lines as atoms in the described molecule; each line provides information about an atom; for instance, its name, its identification number in the molecule…

*What are the expected results of your system?*

From a given initial conformation (defined by the position of every atom composing the molecule), we want to obtain a conformation for this molecule which minimizes its global potential energy.

*How this result can be viewed as a self-organisation result?*

The molecule self-organises because its inner constitutive atoms are going to modify their relative location.

*Which techniques do you use?*

Techniques we use are based on the AMAS theory (in which cooperation is the local criterion that makes components of a system reorganise, see www.irit.fr/SMAC) and involve cooperative agents. These agents are autonomous entities that follow a classical lifecycle (perceive, decide and act) and have a social attitude based on cooperation.

---

[14] Authors : Bernon C., Besse-Patin C., Glize P.

Cooperation is to respect three conditions: (1) all perceived signals must be understood, (2) the information provided must be useful for the agent's reasoning and (3) this reasoning leads to useful actions towards other agents. A cooperative agent knows how to detect situations that infringe these laws and acts to remove them. Those situations are called Non Cooperative Situations (NCS).

*What are the entities inside the system which have to self-organize?*
The atoms have to self-organize in order to get an organization which reflects the spatial conformation we want to obtain.

### Environment Description

*What is the environment of the system composed of?*
The environment is composed of the physical laws that rule the interactions between atoms these laws are static during a session.

### Perturbations Coming from the Environment

*Which kind of perturbations your system can take into account?*
In this case study, there is no perturbation coming from the environment, it is closed, no new agent can appear, for example.

### Entity Description

*If you use agents please describe them in terms of:*
*Their individual goals*
An agent aims at reducing its potential energy.
*Their perceptions*
An agent perceives all atoms in a given area. This area concerns its covalent atoms (strong interactions) and the atoms with which it has non-bonded interactions (weak interactions). We call the agents it perceives, its neighbours.
*Their skills or capabilities (i.e. tasks that an agent can perform in the form of <precondition, action, effect>)*
An agent is able to modify its location within the molecule, it moves. It knows how to evaluate its potential energy. It can evaluate the potential energy of its neighbours, for a given move.
*Their beliefs about other agents or acquaintances*

- The position and the current energy level of other agents in its perception area.
- Their beliefs about themselves
- An agent knows its position, its neighbours, its covalent atoms.
- Their beliefs about the environment of the system (so without the other agents)
- The physical laws concerning the atoms (coming from the environmental database).

### Interaction Description

*Which kind of communication do agents use?*
It is an indirect communication: an agent perceives information about its neighbours. There is no message sending but it could be used.

### Self-Organisation Engine

*Describe the engine (mechanism) of self-organisation processing.*
The engine of self-organisation process is the cooperation (see AMAS theory): an agent tries to reduce its own potential energy and its worst neighbour's energy level (NCS action).

*What elements trigger the re-organisation?*
The fact that an atom wants to reduce its potential energy.
*Is this a local or a global one?*
This is a local criterion. An atom ignores the value of the global energy of the molecule.
*Is this an internal or external one?*
This is internal processing.
*What are the steps (i.e. interaction sequences) involved in the various self-organisation cases?*
For each agent:
*Perceive the potential energy of all its neighbours,*

*Compute its own local potential energy,*

*Spatially move to decrease the worst potential energy while respecting the covalent bonds.*

*When does the self-organisation process stop?*
The process could stop when:

- The variation of the global energy is small (given an epsilon),
- A number of simulation steps is reached,
- Requested by the user.

### 4.4.3. Characterisation of the Self-Organisation Approaches

.

| **Characterisation criteria** | |
|---|---|
| *Is an external control (autonomy) absent ?* | Yes. There is no interaction with the environment data of the problem are given. |
| *Is there a decentralised control?* | Yes, each atom controls the decisions it takes. |
| *Dynamic operation (time evolution)* | Yes, the solution is computed |
| *Endogenous global order* | Yes, the resulting conformation formed by the organisation between atoms comes from their interactions, it is not given. |
| *Emergence from local interactions, emergent properties* | Yes, the very essence of the AMAS theory. The right organization emerges from the reorganisation of the atoms because they try to be satisfied and move when they are not. |
| *Dissipation (energy usage/far-from-equilibrium)* | ?? |
| *Instability (self-reinforcing choices/nonlinearity), Parameter sensibility* | Yes, when an atom modifies its location, others are disturbed. |
| *Multiple equilibriums (many possible attractors)* | Yes. |
| *Criticality (threshold effects/phase changes)* | ?? |
| *Redundancy (insensitivity to damage)* | No. |
| *Adaptation (functionality/tracking of external variations)* | No, in this case study. It could be possible in other problems. |
| *Complexity (multiple concurrent values* | Yes, each agent has its own local function to |

| | |
|---|---|
| *or objectives)* | achieve. |
| *Hierarchies (multiple nested self-organised levels)* | No. |
| *Simple local rules* | Yes, the local rules applied by the agents are very simple. |

## 5. Link with Other Disciplines

### Biology[15]

Biology and Self-Organised Multi-Agent Systems (SOMAS) share common interests as they are equally addressed to complex autonomous and autopoïetic systems. SOMAS will provide powerful modelling tools in biological investigation. There is a distinction between models which represent fully specified systems (these can be formalized with algorithms or implemented as computer programs), and biological theories, which represent less specified intuitions about a process. Modelling is important for a theory by requiring a full specification of the details of that theory in spite of gaps, contradictions and parameters where the process described is weakly known. Moreover, some theories insufficiently specified and so hardly falsifiable (that is the defining characteristic of scientific theories) can use models to test them by comparing performances with empirical observations. It is also an alternative to Occam's razor in establishing the simplicity of a theory through its implementation, not its formulation.

Adaptive MAS are adequate in most of biological modelling, by allowing the development of systems in open, dynamic and underspecified environments, and mainly by exploring system's behaviour exclusively at the micro level of agent's functioning. Effectively, biological systems are the combined result of a huge amount of interactions between autonomous and eventually heterogeneous entities that are sensitive to a more or less constraining environment. At the lower scale, properties of a biologically interesting molecule like DNA or proteins depend on its tridimentional structure. This conformation is the result of interactions (attraction-repulsion) between atoms and between atoms and environment (solvent, electrical gradient, other molecules, etc.). At the cellular scale, the behaviour of the system is much more complex and the dynamic of the macro level (the cell) can be very far from the dynamic of the micro level (the molecular) functioning. In fact, a cell is known to be a very heterogeneous but very structured architecture of molecules (membranes, cytoskeleton, function specific regions, etc.). That is perhaps the biggest challenge of biological modelling. From cells to tissue, from tissues to organism, from organisms to society or ecosystem, and so on, each biological system where global functioning relies on discrete and numerous interactions can be thought as a multi agent system where agents vary in heterogeneity, but are especially characterised by the asymmetric nature of the interactions between them and with their surrounding world: their action in the environment or the society is direct but can go beyond they sensitive range through a collective emergent phenomenon that will indirectly modify the agent state.

**Additional comments on possible collaboration**
*Markus Bongard, University Miguel Hernandez, Alicante, Spain, markus.bongard@umh.es*

---

[15] Author : Mano JP.

Fernandez' group is a group of neurophysiologist thatinvestigate the information processing in the visual system - which represents a working complex, adaptive, dynamical system which displays on various conditions emergent behaviour. The participation on the meeting presented some completely new, "different" points of views to the neurophysiologist and provide new ideas about how MAS research and neurophysiology investigation might lead to some fruitful collaboration:

First, on an abstract level vision science tries to do something which one can see as "reverse engineering" of an existing multi agent system (the visual system). Researcher try to identify the smallest functional unit (which you might call an "agent") and higher "logical" functional units which might be composed of ensembles or even large population of smaller functional units. Because the multi agent system field has apparently developed and defined many theoretical fundamentals and principles (like for example "Holons" mentioned during the meeting), and neurobiology has done much scaffolding in form of axioms and definitions MAS community were looking on during the forum (emergent properties, adaptation, etc.) - this might offer a nice foundation for system theoretical collaboration.

Second, as mentioned previously neurophysiologist are investigating a working, self-organizing, dynamical system which solves for example constantly "real-world" combinatorial optimization problems. Because visual neuroscience has now, by using massive parallel recording techniques, a lot of data which describes this system (and tries to model the system functionality) - all this might serve as basis for collaboration to develop new "bio-inspired" multi agent system or strategies.

## 6. Summary and future work

The SELF-ORG TFG has so far been quite successful in gathering research from various disciplines under the umbrella of self-organisation and emergence.
It is planned to continue work both remotely and in informal meetings organised at regular intervals.
The near future tasks are:

- to elaborate further on the definition of « emergence »
- to complete the list of self-organisation assessment criteria
- to refine and finalise the self-organisation approaches applied on the case studies.
- to reinforce our links with the biology community and to make new links with communities in other fields such as complex systems, economy…
- to update the Web site repository (http://www.irit.fr/TFGSO)
- to propose next Self-organisation in multi-agent systems TFG meeting in Budapest with a modular structure (3 half-days).
- To work towards the preparation of a book covering the work carried out in the TFG rsanging basic concepts, applications, and methods, most likely based on the identified case studies, and intending to serve as a reference.

## 7. A bibliography on emergence and self-organisation

This bibliography is not yet complete but the aim is to enrich it further as the collaborative work of the TFG progresses.

1 Ali, S. M., Zimmer, R.M., (1997) *« The question concerning emergence »,* CASYS'97, Abstract Book, First International Conference on Computing Anticipatory Systems, CHAOS 1997.

2 Atlan, H. (1993) *«Enlightenment to Enlightenment: Intercritique of Science and Myth ,* SUNY Press: Albany, NY.

3 Bak, P. (1996) *«How Nature Works: The Science of Self-organized Criticality»,* Springer-Verlag: New York.

4 Belew, R.K. & Mitchell, M. (1996) Adaptive Individuals in Evolving Populations: Models and Algorithms*»,* Addison Wesley: Reading, Mass.

5 Bernon C., Camps V., Gleizes M-P., Picard G., (2004), *«Tools for Self-Organizing Applications Engineering»,* Engineering Self-Organising Systems, Nature-Inspired Approaches to Software Engineering, LNAI 2977 State of the Art Survey, Springer Verlag 2004, Di Marzo Serugendo G., Karageorgos A., Rana O.F., Zambonelli F. Editors pages 283-298

6 Bourjot C., Chevrier V., Thomas V., (2002) *« How social spiders inspired an approach for region detection»* In AAMAS 2002, First International Joint Conference on Autonomous Agents and Multi-Agent Systems, Bologne, pp426-433, Juillet 02.

7 Bourjot, C., Chevrier, V., Thomas, V., *« A new swarm mechanism based on social spiders colonies: from web weaving to region detection »,* Web Intelligence and Agent Systems: An International Journal, 1, 1, pages 47-64.

8 Bourjot C., Chevrier V., (2004) *«A Platform for the analysis of artificial self-organized systems»,* In 2004 IEEE International Conference on Advances in Intelligent Systems - Theory and Applications - AISTA'2004.

9 Brueckner S.A., Parunak H. V.D., (2004) *« Self-Organizing MANET Management »*, Di Marzo Serugendo G., Karageorgos A., Rana O.F., Zambonelli F. Editors of The First International Workshop on Engineering Self-Organising Applications (ESOA) Melbourne, Australia, July 2003 LNAI 2977 State of the Art Survey Springer Verlag 2004

10 Van Brussel H., Wyns J., Valckenaers P., Bongaerts L., Peeters P., (1998) *«Reference Architecture for Holonic Manufacturing Systems: PROSA»*, Computers in Industry, Special Issue on Intelligent manufacturing Systems Vol37, n°3 pp255-276 1998.

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

12 Capera D., Gleizes M-P., Glize P., (2004) *« Mechanism Type Synthesis based on Self-Assembling Agents »,* Journal of Applied Intelligence Artificial 2004

13 Capera D., Georgé J-P., Gleizes M-P., Glize P. (2003), *« Emergence of organisations, emergence of functions»,* Proceedings of the AISB'03 symposium on Adaptive Agents and Multi-Agent Systems, University of Wales, Aberystwyth, 7-11 April 2003 , pages 103 - 108

14 Capera D., Gleizes M-P., Glize P. (2004), *«Self-Organizing Agents for Mechanical Synthesis»,* Engineering Self-Organising Systems Nature-Inspired Approaches to Software Engineering, LNAI 2977 State of the Art Survey, Springer Verlag 2004, Di Marzo Serugendo G., Karageorgos A., Rana O.F., Zambonelli F. Editors, pages 169-185.

15 Capera D., Georgé J-P., Gleizes M-P., Glize P. (2003*), « The AMAS theory for complex problem solving based on self-organizing cooperative agents »* 1st

International workshop on Theory And Practice of Open Computational Systems TAPOCS, IEEE 12th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises WETICE 2003, Johannes Kepler Universitaet, Linz – Austria, 2003.06.09 – 11

16 Casti, J. (1994) Complexification: Explaining a Paradoxical World through the Science of Surprise. Addison Wesley: Reading, Mass.

17 Cowan, G.A, Pines, D. & Meitzer, D. (1994) Complexity, Metaphors, Models, and Reality. Addison Wesley: Reading, Mass.

18 Darwin, C. (1859) On the Origin of Species by Means of Natural Selection. Murray: London.

19 Di Marzo Serugendo G., Karageorgos A., Rana O.F., Zambonelli F., (2003) *« The First International Workshop on Engineering Self-Organising Applications (ESOA »,* Melbourne, Australia, July 2003 LNAI 2977 State of the Art Survey Springer Verlag 2004

20 Forrest, S., (1990) *« Emergent computation: self-organizing, collective, and cooperative phenomena in natural and artificial computing networks »,* Proceedings of the Ninth annual CLNS conference, 1990.

21 Foukia N., Hassas S., (2004) *« Managing Computer Networks Security through Self-Organization : A complex System Perspectives »,* The First International Workshop on Engineering Self-Organising Applications (ESOA), Di Marzo Serugendo G., Karageorgos A., Rana O.F., Zambonelli F. Editors, Melbourne, Australia, July 2003 LNAI 2977 State of the Art Survey Springer Verlag 2004

22 Gardner, M. (1970) *« Mathematical games: the fantastic combinations of John Conway's new solitaire game "Life"* ». Scientific American (October 1970): 120-123.

23 Gechter F., Chevrier V., Charpillet F. (2004) *«Localizing and Tracking Targets with a Reactive Multi-Agent System».* In Second European Workshop on Multi-Agent Systems - EUMAS'04. (Barcelona, Spain), pp255-262, 2004

24 Gechter F., Chevrier V., Charpillet F. (2004) *« Localizing and Tracking Targets with a Reactive Multi-Agent System».* In Third International Joint Conference on Autonomous Agents and Multi-Agent Systems - AAMAS'04. (New York, USA), pp1488-1489, 2004.

25 Gechter F., Chevrier V., Charpillet F. (2004) *« A Reactive Multi-Agent System for Localization and Tracking in Mobile Robotics»,* 16th IEEE International Conference on Tools with Artificial Intelligence - ICTAI'2004 (Florida, USA). 2004

26 Gell-Mann, M. (1995*), « What is complexity? »* Complexity 1: 16-19.

27 Georgé J.P., Edmonds B. Glize P., (2004) *"Making self-organizing adaptive multi-agent systems work"* F. Bergenti, M-P. Gleizes, and F. Zambonelli, Editors, Methodologies and Software Engineering for Agent Systems. The Agent-Oriented Software Engineering Handbook. Kluwer Publishing, 1-4020-8057-3, 2004.

28 Goldstein, J., (1999) *« Emergence as a Construct : History and issues »,* Emergence Volume 1, Issue 1, pages 49-71, 1999

29 HilaireV., Koukam A., and Gruer P. (2002) *« A mechanism for dynamic role playing . »* In Agent Technologies, Infrastructures, Tools and Applications for E-Services, number 2592 in Lecture Notes in Artificial Intelligence. Springer Verlag, 2002

30 Holland, J. (1995) *«Hidden Order: How Adaptation Builds Complexity».* Addison Wesley: Reading, Mass.

31 Holland, J. (1998) *Emergence: From Chaos to Order*. Addison Wesley: Reading. Mass.

32 Kauffman, S.A. (1993) *«The Origin of Order: Self-Organization and Selection in Evolution».* Oxford University Press: New York.

33 Koestler A. (1967) *« The Ghost in the machine. »* Huchinson, 1967.

34 Mamei M., Zambonelli F., (2004) *« Self-Organization in Multi-Agent Systems : A Middleware Approach »,* First International Workshop on Engineering Self-Organising Applications (ESOA), Di Marzo Serugendo G., Karageorgos A., Rana O. F., Zambonelli F. Editors, Melbourne, Australia, July 2003 LNAI 2977 State of the Art Survey Springer Verlag 2004

35 Maturana, H.R. & Varela, F.J. (1980) Autopoiesis and Cognition: The Realization of Living. Reidel: Boston.

36 Nagpal R., *« A Catalog of Biologically-Inspired Primitives for Engineering Self-Organization »,* (2004) The First International Workshop on Engineering Self-Organising Applications (ESOA), Di Marzo Serugendo G., Karageorgos A., Rana O. F., Zambonelli F. Editors, Melbourne, Australia, July 2003 LNAI 2977 State of the Art Survey Springer Verlag 2004

37 Parunak H. V.D., *« Go to the Ant: Engineering Principles from Natural Multi-Agent Systems »,* Annals of Operations Research 75 69-101 Special Issue on Artificial Intelligence and Management Science

38 Parunak H. Van Dyke, Vanderbok R.S., (1997) *« Managing emergent behavior in distributed control systems »,* Proceedings of ISA-Tech'97

39 Parunak, H. Van D., Brueckner, S., (2001) *« Entropy and self-organization in multi-agent systems »,* Proceedings of Autonomous Agents'01, p. 124-130, 2001.

40 Rodriguez S., Hilaire V. and Koukam A. (2003) *« Towards a methodological framework for holonic multi-agent systems . »* In Fourth International Workshop of Engineering Societies in the Agents World, Imperial College London, UK (EU), 29-31 October 2003.

41 Rodriguez S., Hilaire V., and Koukam A. (2005) *« Formal specification of holonic multiagent system framework. »* In Intelligent Agents in Computing Systems - The Agent Days in Atlanta, Lecture Notes in Computer Science. Springer-Verlag, 2005. to appear.

42 Servat, D., Leonard, J., Perrier, E., Treuil, J.P., (1999) *« The Rivage Project : a new approach for simulating runoff dynamics »,* Modelling of transport processes in soils, 24-26 Novembre 1999, Leuven, J.Feyen & K.Wiyo.Eds, Wageningen Pers, The Netherlands, pages 592-601

43 Steels, L., *« The Spontaneous Self-organization of an Adaptive Language »,* Koichi Furukawa and Donald Michie and Stephen Muggleton, editors, Machine Intelligence 15, pages 205—224, St. Catherine's College, Oxford: Oxford University Press.

44 Thomas V., Bourjot C., Chevrier V., Desor. D., (2002), *« MAS and RATS Multi-agent simulation of social differentiation in rats_ groups. Interest for the understanding of a complex biological phenomenon»,* International Workshop on Self-Organization and Evolution of Social Behaviour, Monte Verita, September 8-13, 2002.

45 Thomas V., Bourjot C., Chevrier V., Desor. D., (2002), *« Hamelin: A model for collective adaptation based on internal stimuli»,* In Stefan Schaal, Auke Ijspeert, Aude Billard, Sethu Vijayakumar, John Hallam, and Jean-Arcady Meyer, editors, From animal to animats 8- Eighth International Conference on the Simulation of Adaptive Behaviour 2004 - SAB_04, Los Angeles, USA, pages 425-434, Jul 2004

46 De Wolf T., Samaey G., Holvoet T., and Roose D., (2005) « Decentralised *autonomic computing: Analysing self-organising emergent behaviour using*

*advanced numerical methods »,* Proceedings of the Second International Conference on Autonomic Computing, IEEE Computer Society Proceedings, pp. 1-12, 2005

47 De Wolf T., and Holvoet T., (2005) *« Emergence versus self-organisation: different concepts but promising when combined »,* Engineering Self Organising Systems: Methodologies and Applications (Brueckner, S. and Di Marzo Serugendo, G. and Karageorgos, A. and Nagpal, R., eds.), vol 3464, Lecture Notes in Artificial Intelligence, 2005

48 Zurek, W.H. (ed.) (1990) *« Complexity, Entropy and the Physics of Information».* Santa Fe Institute Studies, vol.8, Addison Wesley: Reading, Mass.