

Self-Organisation and Emergence in MAS: An Overview

Giovanna Di Marzo Serugendo
University of Geneva, Switzerland
E-mail: Giovanna.Dimarzo@cui.unige.ch

Marie-Pierre Gleizes
IRIT, Université Paul Sabatier, France
E-mail: Marie-Pierre.Gleizes@irit.fr

Anthony Karageorgos
University of Thessaly, Greece
E-mail: karageorgos@computer.org

Keywords: multi-agent systems, self-organisation, emergence

Received: June 5, 2005

The spread of the Internet and the evolution of mobile communication, have created new possibilities for software applications such as ubiquitous computing, dynamic supply chains and medical home care. Such systems need to operate in dynamic, heterogeneous environments and face the challenge of handling frequently changing requirements; therefore they must be flexible, robust and capable of adapting to the circumstances. It is widely believed that multi-agent systems coordinated by self-organisation and emergence mechanisms are an effective way to design these systems. This paper aims to define the concepts of self-organisation and emergence and to provide a state of the art survey about the different classes of self-organisation mechanisms applied in the multi-agent systems domain. Furthermore, the strengths and limits of these approaches are examined and research issues are provided.

Povzetek: Članek opisuje pregled samoorganizacije v MAS.

1 Introduction

Natural self-organising systems function without central control and operate based on contextual local interactions. The particularity of self-organised systems is their capacity to spontaneously (without external control) produce a new organisation in case of environmental changes. These systems are particularly robust, because they adapt to these changes, and are able to ensure their own survivability. In some cases, self-organisation is coupled with emergent behaviour, in the sense that although individual components carry out a simple task, as a whole they are able to carry out complex tasks emerging in a coherent way through the local interactions of the various components.

The complexity of today's applications is such, e.g. world scale, that no centralised or hierarchical control is possible. In other cases, it is the unforeseeable context, in which the application evolves or moves, which makes any supervision difficult. Therefore, we are witnessing an increased interest from both the academic community and the industry in naturally inspired (robust and simple) solutions for building modern applications favouring self-organisation and/or emergence of properties.

We can foresee that among the applications of tomorrow, a great many of them will be biologically inspired: self-organising sensors networks, allowing the control of aerospace vehicles, or of dangerous zones;

self-organising traffic management, allowing re-routing of emergency vehicles, or individual cars; storage facilities, or self-managing operating systems facilities. Some others applications tackle with complex problem solving in which complexity is due to the great space search such as optimisation problems and non linear problems.

Software agents naturally play the role of autonomous entities subject to self-organise themselves. Usually agents are used for simulating self-organising systems, in order to better understand or establish models. The tendency is now to shift the role of agents from simulation to the development of distributed systems where components are software agents that once deployed in a given environment self-organise and work in a decentralised manner towards the realisation of a given (global) possibly emergent functionality.

Sections 2 and 3 review the notions of self-organisation and emergence respectively. Section 4 provides the description of several implementations in MAS. Section 5 discusses the strengths and limits of self-organising approaches. The main problems and challenges related to the software engineering of self-organising systems which exhibit emergent properties are discussed in Section 6. Finally, Section 7 concludes the paper.

2 Self-Organisation

2.1 History

By studying the social behaviour of insects (termites), Grassé [29] proposed in 1959 the theory of *stigmergy*, which can be summarised in “*the work excites the workers*”. The consequence is that direct interactions are not necessary to coordinate a group, for example indirect communications through environment are enough. Coordination and regulation tasks are realised on the basis of information deposited into the environment, without central control. In the case of ants and termites, stigmergy is ensured by depositing a chemical substance in the environment, called *pheromone*.

In the 70es, the term self-organisation itself has been established by Nobel Prize Ilya Prigogine [26] and his colleagues through thermodynamics studies. Essentially the idea is that open systems decrease their entropy (order comes out of disorder) when an external energy is applied on the system. Matter organises itself under this external pressure to reach a new state where entropy has decreased. Compared to the stigmergy concept identified by Grassé, there is a fundamental difference here. Indeed, in the first case self-organisation results from a behaviour occurring from inside the system (from the ants or termites themselves). In the second case, self-organisation is the result of a pressure applied from the outside on the system.

In the 70es, through biological studies, Francisco Varela [61] established the notion of autopoiesis (meaning self-production) as being the process through which an organisation is able to produce itself. Autopoiesis applies to closed systems made of autonomous components whose interactions self-maintain the system through the generation of system’s components, such as living systems (cells, or organisms).

Koestler [37] in the late 60es established the definition of holons and holarchies. Holons are at the same time whole systems and parts of larger systems. Holarchies are hierarchies of such holons. Koestler gives a hierarchical view of self-organisation, which applies to the universe or to enterprise organisations. The idea here is that, for complex organisations, order appears from disorder, due to simple relations that statistically evolve through complex relations progressively organising themselves.

During the last 20 years, research in artificial systems has been oriented towards introducing self-organisation mechanisms specifically for software applications. These different works take diverse inspiration: from stigmergy, to autopoiesis, or to the holon concept. Recently, in addition to reproducing natural system behaviour into artificial systems, recent research efforts have been oriented towards introducing self-organisation mechanisms specifically for software applications [20]. Section 4 describes such mechanisms in more details.

2.2 Examples

Natural self-organising systems include well-known examples concerning social insects, such as ants, termites and honey bees. Communication occurs through stigmergy by the means of pheromone deposited into their environment. Other collective behaviours of animals referred to as self-organising are flocks of birds, and schools of fish. By following simple rules, such as getting close to a similar bird (or fish) but not too much, getting away from dissimilar birds (or fishes), they are able to collectively avoid predators.

Social behaviour of humans is also self-organised and gives rise to emergent complex global behaviours. Human beings typically work with local information and through local direct or indirect interactions producing complex societies.

Biology provides a great source of self-organising systems as well. Examples include the immune system of mammals, the regeneration of cells and brain behaviour.

Among artificial multi-agent based self-organising systems, we observe different trends ranging from application of naturally-inspired self-organising models, to the establishment of new mechanisms and whole infrastructures supporting self-organisation of artificial systems. Swarms provide a great source of inspiration, especially for fixed and mobile networks systems management [11], such as routing, load balancing [43], or security [25]. Holarchies as well have inspired researchers dealing with e-Government and e-Society issues [60]. At the level of whole infrastructures (middleware) supporting artificial self-organising systems, some works take their inspiration from magnetic fields [40], or ants [2].

2.3 Definition

Self-organisation essentially refers to a spontaneous, dynamically produced (re-)organisation. We present here several definitions corresponding to the different self-organisation behaviours identified in Section 2.1.

Swarm Intelligence. According to [Bonabeau, 1999] mechanisms identifying swarms behaviour are: 1. Multiple interactions among the individuals; 2. Retroactive positive feedback (increase of pheromone when food is detected); 3. Retroactive negative feedback (pheromone evaporation); 4. Increase of behaviour modification (increase of pheromone when new path is found).

Decrease of entropy. Prigogine and his colleagues have identified four necessary requirements for systems exhibiting a self-organising behaviour under external pressure [26]. “Mutual Causality: At least two components of the system have a circular relationship, each influencing the other. Autocatalysis: At least one of the components is causally influenced by another component, resulting in its own increase. Far-from equilibrium condition: the system imports a large amount of energy from outside the system, uses the energy to help renew its own structures (autopoiesis), and dissipates rather than accumulates, the accruing disorder

(entropy) back into the environment. Morphogenetic changes: At least one of the components of the system must be open to external random variations from outside the system. A system exhibits morphogenetic change when the components of the system are themselves changed [15].”

Autopoiesis. “An autopoietic system is organised (defined as a unity) as a network of processes of production (transformation and destruction) of components that produces the components that: 1. Through their interactions and transformations continuously regenerate and realise the network of processes (relations) that produced them; and 2. Constitute it (the machine) as a concrete unity in the space which they [the components] exist by specifying the topological domain of its realisation as such a network” [61].

Artificial Systems. Works of the Agentlink Technical Forum on Self-Organisation in MAS [21] have established two definitions of self-organising systems: 1. "Strong self-organising systems are systems that change their organisation without any explicit, internal or external, central control"; 2. "Weak self-organising systems are systems where reorganisation occurs as a result of internal central control or planning".

Furthermore, self-organisation implies organisation, which in turn implies some ordered structure and component behaviour. In this respect, the process of self-organisation changes the respective structure and behaviour and a new distinct organisation is self-produced.

When self-organisation meets emergence. Emergence is the fact that a structure, not explicitly represented at a lower level, appears at a higher level. In the case of dynamic self-organising systems, with decentralised control and local interactions, intimately linked with self-organisation is the notion of emergent properties. The ants actually establish the shortest path between the nest and the source of food. However in the general case, as pointed out by [18] self-organisation can be witnessed without emergence and vice-versa.

3 The Emergence Concept

3.1 History

The emergent phenomena are studied since the Greek antiquity and can be found in the writings of the Socrate periods with the notion of “the whole before the parts” or “the whole is more than all the parts”. There were two different schools for studying the emergence: the proto-emergentism during the XIX century and the neo-emergentism during the XX century.

The proto-emergentists consider the emergent process as a black box (see Figure 1). Only the inputs and the outputs at the lowest level can be discerned. We don’t know how the entries are transformed in outputs. Researchers such as: G.H. Lewes, C.L. Morgan, J.S. Mill, S. Alexander, D. Broad, W. Wheeler, and A.N.

Whitehead try to explicit the characteristics of emergent phenomena.

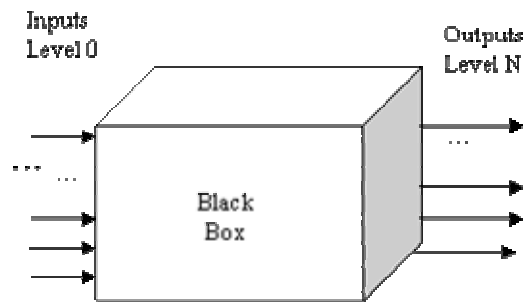


Figure 1: Proto-emergentist view

From 1930 until just now, a different perspective has been envisaged, by a movement called the neo-emergentism. It has its root in dynamic of systems in Physics, in Mathematics and in Computer Science with main examples being the work of Haken, Holland, Kauffman, Langton, Prigogine, and Thom. Its aim is to develop tools, methods and constructions which enable the expression of the emergent process as less dense and by consequence as less miraculous (Figure 2). This movement tries to understand and to reproduce the process which leads to emergence.

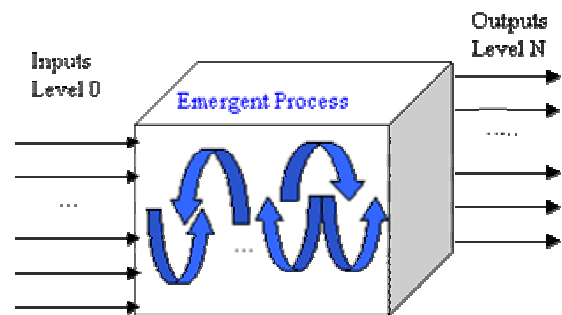


Figure 2: Neo-emergentist view

3.2 Examples

To illustrate the notion of emergent phenomena, this section presents examples of systems where emergent phenomena can be observed.

The first example is taken from natural systems and concerns foraging ants [16]. A foraging ant has the role to explore an environment to find food. When it finds food it comes back to the nest in tracing the path in the environment with pheromone. The shortest path to find food is the structure which emerges from the collective activity of the ants. This path has reality only for an observer of the system and an ant does not view it.

Another example concerns an application where robots have to transport boxes from always the same departure room to a destination one. Two corridors are available to go from one room to the other and two robots cannot cross in a corridor and there is no sense

associated to them. The robots have a local perception. In using cooperative attitude to embody robots, we can observe corridors dedication and traffic way [50].

The apparition of conscience is an example of emergent phenomenon for humans. The conscience is viewed by Searle [57] as a property of the brain at the higher or global level. Biologically, the brain is a complex system composed of a set of neurons and interactions between them. These neurons are the lower or micro level. Nowadays, we cannot understand or explain the conscience in observing the neurons and their interactions.

3.3 Definition

The emergence is a captivating concept and we try to explain it in answering the following questions:

- What does emerge?
- What are the characteristics of an emergent phenomenon? These characteristics must enable to answer yes or no to the question: “is this phenomenon an emergent one?”
- What are the properties of a system producing emergent phenomena? These properties can guide designers to build systems which provide emergent phenomenon.
- What can be emergence in artificial systems? How can you decide if a program provides an emergent result or not?

The object of emergence is often called phenomenon and it can be a structure or a framework such as the Bénard’s cells, a behaviour such as the glider in the game of life [4], or a function (not as mathematic function but as the functionality of a system) such as the building of a course schedule by several local entities [44], [52].

An emergent phenomenon requires at least two levels (a micro and a macro level), and needs to be observable at least at the macro level. Its main property is the irreducibility of the properties of a high level theory to properties of a lower level theory [1]. In general, there are interdependencies between the levels, the macro level constrains the micro level and the micro level causes the macro level. The phenomenon must show novelty: something new is produced that did not exist previously; must be ostensible; and must produce some coherence in the sense that it has its own identity but it is strongly linked to parts that produce it [28]. A chain of linear activities enables explanation and predictability of a collective phenomenon. On the opposite, an emergent one needs non linear activities at the micro-level. For a given phenomenon, if most of the previous properties can be observed then the phenomenon can be qualified as emergent.

Engineers should be provided with a guide including models, tools and methods to design systems having an emergent behaviour or presenting emergent results. Furthermore, the guide should list the main properties such systems should have. To provide emergent phenomena, a system or a mechanism must at least have two levels. The system must present a dynamic during its time life. 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 [36], which is always far from equilibrium. Near these equilibriums, a system has the ability to self-organise allowing an emergent phenomenon.

The emergence in artificial system is conceptually close to emergent computation defined by Stephanie Forrest [24] as follows:

- a collection of interactive agents : the process;
- an epiphenomenon produced by this process at the macro level;
- a natural interpretation of this epiphenomenon as computation or computation results.

An operational definition is given by the SMAC team at IRIT [13]. This “technical” definition of emergence has strong computer science coloration and it is based on two points:

1. **The subject.** The goal of a computational system is to realise an adequate function, judged by a relevant user. It is this function, which may evolve during time, that has to emerge.
2. **The condition.** This function is emergent if the coding of the system does not depend in any way of the knowledge of this function. This coding has to contain the mechanisms allowing the adaptation of the system during its coupling with the environment, so as to tend anytime towards the adequate function.

Therefore, when we design an agent for a multi-agent system, the code of the agent doesn’t contain any knowledge of the collective function we want the MAS to compute. As a result, no agent controls the global system.

4 Implementation in MAS

Studies on self-organisation and emergence focus on naturally inspired approaches (bio-inspired approaches [41], and socially-based approaches [30]) and non naturally inspired approaches. Researchers have been experimented with several mechanisms leading to self-organisation and often at the same time to emergent phenomenon on different kinds of applications [5]. The different approaches can be divided in five classes depending on the mechanisms they are based on:

- direct interactions between agents using basic principles such as broadcast and localisation;
- indirect interactions between agents and stigmergy;
- reinforcement of agent behaviours;
- cooperation behaviour of individual agents;
- choice of a generic architecture.

For a more general survey of languages and platforms for MAS implementations not directly related to self-

organising mechanisms, the interested reader can refer to [9].

4.1 Mechanisms based on direct interactions

Zambonelli et al. [64] discuss different ways to engineer self-organisation. The approaches proposed consist in using few basic principles, such as localisation and broadcast, coupled with local interactions and local computations done by agents in order to provide a final coherent global state. These algorithms differ from traditional distributed algorithms in that they focus on ensuring that they eventually will converge to and maintain a desired stable state despite micro-level contingencies and any perturbations in the environment, for example changes in the network structure.

Typical examples of such mechanisms are those applied in the areas of self-assembly and distributed self-localisation where the formation of regular spatial patterns in mobile objects is required. An example is described in [40] where simple leader election algorithm determines the centre of gravity of the objects and propagates it to all objects which move until a specific distance from the centre is reached. The result eventually is a circular organisation of objects. The same mechanism is used in the system for modelling fluid dynamics [58]. Local interactions between drops and interactions with a physical environment enable the formation of rivers or ponds.

These mechanisms focus on changing the structural aspects of the agent organisation, such as topological placement of agents and agent communication lines.

4.2 Mechanisms based on stigmergy

The self-organisation mechanisms based on the stigmergy concept aim at achieving complex system behaviours resulting of indirect interactions between agents. These interactions are due to changes in the environment. This behaviour leads towards the desired global system behaviour.

Recently, several approaches to self-organisation relying on this idea of stigmergy have been proposed and their effectiveness in achieving difficult global coordination tasks has been demonstrated. For instance, this mechanism has been used for manufacturing control [35], supply network management [55], managing computer networks security [25] and coordination of unmanned vehicles [48]. Stigmergy has also been implemented with social spiders to detect regions in a scene [10]. This principle is also used to obtain the formation of non-symmetric patterns in self-assembly applications [40] which in some cases are not exactly known in advance but emerge during system execution [53]. An example of such non-symmetrical pattern formation using principles of biological formation of morphogenesis is given in [40].

These mechanisms can be evaluated by experimentation, for example by simulation and prototyping [23], [38]. In particular there is a tendency to integrate simulation experiments in the methodologies for engineering such systems, such as the one described

in [47]. In such approaches, the design phase involves selecting an appropriate self-organising model and verifying its correctness via experimentation. Such a model may be relevant, but not necessarily the most suitable for the particular application scenario. Therefore, the model is calibrated via iterative refinement based on the experimentation results.

In these cases, due to the non-linearity and the complexity of the phenomena involved, neither it is possible to have direct control of the system behaviour nor can it be proven that the desired behaviour will be achieved. Furthermore, the resulting system state cannot be accurately known in advance and multiple solutions can be reached. One can only obtain some statistical confidence about the system convergence to the desired globally coordinated behaviour with experimentation.

4.3 Mechanisms based on reinforcement

In some approaches self-organisation is based on the capabilities of the agents to modify dynamically their behaviour according to some reinforcement. It consists in the following basic principles: rewards increase agent behaviour and punishments decrease agent behaviour. The consequence is that an individual agent can adapt its capabilities and we can observe specialisation of roles for example. In these approaches self-organisation is based on adaptive behaviour capabilities of individual agents which are dependent on particular agent architectures. In these approaches, agents dynamically select a new behaviour (or action) based on the calculation of a probability value which is dependent on the current agent state and the perceived state of the environment, as well as on the quality of the previous adaptation decisions, for example the ones discussed in [39] and [17]. Other early approaches to self-organisation that re-assign roles and responsibilities to different organisational nodes are detailed in [49].

A typical example of this approach is the model of adaptive agents described in [62]. The model focuses on dynamically adapting logical relations between different behaviours, represented by roles, an agent can successively follow starting from its current state. These relations are used to select the new agent behaviour when adaptation of behaviour needs to be made. Agent behaviour is described as a graph termed *behaviour graph*. A behaviour graph includes two types of nodes corresponding to *roles* and *links*. Role nodes are connected to each other only via appropriate link nodes, which contain *conditions* specifying when the agent can switch between the respective roles. Adaptive role selection takes place on runtime based on *factors* associated with the links of the behaviour graph. Factors are parameters representing properties of agents and their perceived environment whose values can change dynamically during agent execution.

4.4 Mechanisms based on cooperation

The *Organisation Self-Design* (OSD) framework [33] uses the primitives of agents composition and decomposition. Decomposition involves division of an

agent into two and can be performed to respond to overwhelming environmental demands. Composition merges two agents into one and can be useful when communication overheads between the two agents are too high. The system tries to be cooperative with its environment in creating one agent or in merging two agents in order to improve the response time to the environment. The initial organisation starts with one agent containing all domain and organisational knowledge. Simulation results demonstrate the effectiveness of the approach in adapting to changing environmental demands.

Cooperation is also used in the AMAS theory [27] where the desired collective behaviour emerges, and can always occur as the result of cooperation [13], [14]. This emergent outcome corresponds to the delivered system functionality (referred to as the *global function*), which is only modelled using emergence; in other words there is no agent having a global view of the system status or purpose and no centralised control. Each agent possesses the ability of self-organisation, for example the capability to locally rearrange its interactions with other agents and the environment depending on its knowledge, on its representation of the others and on the individual task it has to solve. This enables realising dynamic changes in the global system function without explicitly coding the modifications at the upper level of the system. Self-organisation is founded on the capability agents possess to be locally “cooperative”. Cooperation capabilities do not imply that agents are always helpful or altruistic but they are able to recognise cooperation failures called *Non Cooperative Situations (NCS)* (which correspond to exceptions found in classical programs) and handle them. The local handling of NCS maximises the flexibility and adaptation capability of the system to unexpected situation occurring due to the dynamism of the agent interactions and the environment.

4.5 Mechanisms based on generic architecture

A particular class of self-organisation mechanisms is based on generic reference architectures or meta-models of the agents’ organisation which are instantiated and subsequently dynamically modified as needed according to the requirements of the particular application.

Examples of reference architectures are the mediator architecture proposed by Maturana and Norrie [42] and the PROSA [8] architecture which are both based on the holonic hierarchy model. The holonic hierarchy model involves structural patterns that form nested hierarchies of self-replicating structures named *holarchies* [37]. The elements of holonic systems are referred to with the term *holon* which is a combination of the Greek word *holos*, meaning “whole”, with the suffix “on” meaning part as in proton or neuron.

A common aspect in reference architectures is that they involve characteristic agent types from which the basic agents of a holonic organisation are derived. For example, the mediator reference architecture is based on

the mediator agent type. In PROSA [8] the holonic organisation consists out of three types of basic holons — order holons, product holons, and resource holons. When agents are organised according to the holonic metaphor they participate in holons forming holonic structures. Self-organisation then refers to altering the holonic hierarchy following perturbations of the agent environment using a known decision making technique such as fuzzy-evolutionary reasoning [59].

Examples of approaches based on meta-models and architectural reflection are presented in [22] and [54]. In such approaches, the current system architecture organisation is described as a particular configuration of a generic architectural meta-model which provides the architectural components and their features and also an associated set of architectural constraints that define how and when to safely reconfigure the software architecture.

The meta-model configuration can be inspected and modified at run-time. Modifications of the architecture meta-model result in modifications of the software architecture itself, and the architecture is therefore reflective. Such dynamic modifications can take place either automatically, as is the case in [22] or after user intervention as is done in [54]. The common technique for representing such architectural meta-models is as a typed, directed configuration graph.

5 Strengths and Limits

Mechanisms based on direct interactions have the significant advantage that they enable the design of specific robust self-organised behaviours with exactly known outcomes. However, as mentioned in [64] these approaches are needed only to a limited number of applications. The reason is that only simple global equilibrium states (or patterns of activity) that can be modelled in simple linear terms can be achieved. As a result when more complex behaviour involving non-linear interactions is needed then either too many restrictions for the system operation need to be made or direct mechanisms cannot be applied.

The mechanisms based on stigmergy have additional advantages. Firstly, they enable increased reusability since they make possible to reuse the strengths of known self-organisation mechanisms from biology to build self-organising software. Secondly, once modelling and experimentation for the purposes of calibration has been carried out, the simulation models can be the basis for the actual implementation, reducing thus development time and resources required and hence facilitating development. Furthermore, the simple local behaviours they are based on are quite easy to implement, resulting in increased ease of programming. Furthermore, the multi-solution capability of these mechanisms is one of their strengths since it increases their robustness. Furthermore, although suboptimal solutions are more likely to occur, the effectiveness of these mechanisms is relatively high compared to their low development cost.

The mechanisms based on cooperation behaviour, enable to treat applications with continuous or discontinuous global behaviour. The bottom-up design simplifies also the development and the resulting systems

are robust, because adaptive. For instance, the AMAS theory guarantees that the system only adapts its behaviour to be cooperative with its environment and to satisfy it. The difficulty lies in the exhaustive list of all the non cooperative situations an agent can be faced on. Nevertheless, this is always theoretically feasible because the number of non cooperative situations related to the agent skills is enumerable.

However there are also disadvantages which are essentially related with harnessing emergent behaviour. Firstly, it is currently not possible to effectively control the behaviour of such systems. As a result it is common for undesired emergence states to occur [47]. Furthermore, there can be cases where specific global states are required to emerge, such as the positions of robot players in a football game and hence the many possible solutions offered by such mechanisms can be a problem. A relevant case is when a global solution has emerged and then it is only desirable to maintain it via self-organisation and not converging to another one.

The rest of the mechanisms have similar limitations. An additional strength of mechanisms based on adaptive architectures and meta-mechanisms is that modelling is done using agent-oriented software engineering terms which increases ease of understanding by software designers – in contrast to applying a model from another discipline which would require them to obtain the necessary knowledge to master the terminology and the concepts involved. However this comes to the expense of increased difficulty in modelling global emergent behaviour.

6 Problems and Challenges

From a multi-agent systems development point of view, the central question is: how to program single agents so that, when taken as a whole, they self-organise. In the particular case of multi-agent systems, the interest and the difficulty lies in having both self-organisation and emergent properties, mainly emergent functionality that arises from individual simple tasks performed by the agents. Therefore, the engineering of self-organising applications needs means to define a global goal, and to design local behaviours so that the global behaviour emerges. This is difficult, because the global goal is *not* predictable as the sum or a function of the local goals. Consequently, the verification task turns out to be an arduous exercise, if not realised through simulation.

Traditional software engineering techniques are insufficient, since they are based on interfaces fixed at design time, or well established ontology. As for current methodologies, they only make it possible to define a global behaviour when it is a function of the behaviour of the various parts.

Traditional practices in multi-agent systems introduce basic techniques for autonomously interacting or retrieving information, such as agent coordination, service description, or ontology [6]. However, these techniques rely on pre-programmed interaction patterns, preventing adaptation to unexpected environmental changes. Current engineering practices, which directly address self-organisation, consist in designing distributed algorithms taking inspiration from natural mechanisms,

both bio-inspired and socially-inspired. Some agent-oriented methodologies such as ADELFE [51] provide to designer means to design self-organising systems. More recently, specific electronic interaction mechanisms, non-naturally inspired, are being defined, and middleware technology developed, that will help the development of self-organising applications. However, verification and whole engineering methods remain open issues.

Currently, it is necessary to find means to “control” emergence to use it to solve problems. It is antinomic to speak about emergence and about control on the emergence. But, when designing artificial systems, it is necessary to have operational definition and tools to enable such systems to produce the wanted emergent phenomenon.

In addition the environment plays an important role both as a coordination media and as source of changes and adaptation for the agents. The environment, its engineering and its role in self-organising systems must be well understood and not be underestimated. For a deeper discussion on environments, the interested reader may refer to [63].

A research axis will be to find new principles, theories, models, mechanisms and methodologies to engineer self-organising systems with or without emergent phenomena. In this perspective it is important to be aware of the differences, and to distinguish solutions that tackle self-organisation issues only (without intended causal emergence); emergent issues only (without self-organisation), and solutions that intend to consider both cases in the resulting system. However, in any cases, this is a delicate problem, in the sense that unintended emergent phenomenon that have a causal effect on the system may always arise.

The growing complexity of applications needs solutions that favour autonomous, robust and adaptive systems. Natural systems must be an inspiration sources but we have to devise really new techniques, mechanisms to design self-organisation and emergent phenomenon. This new wave of systems can be called neo-computation and will be useful for designing applications in the domains such as autonomic computing, pervasive and ubiquitous computing.

7 Conclusion

Self-organisation and emergence interest more and more the community of computer scientists and in particular the MAS developers. This craze is due to the fact that self-organisation enables to tackle a new field of applications and that multi-agent systems are well adapted to implement self-organisation.

The paper aims are twofold: it clarifies these two concepts and proposes operational definitions; it then gives an overview of researches on self-organising MAS and emergent phenomena produced by MAS. The different mechanisms studied can be grouped into five families: direct mechanisms, characterized by simple principle of functioning in the agents and direct

communication; mechanisms based on stigmergy, which use indirect interactions between agents and where the perceptions reinforce some agent actions; reinforcement mechanisms, which enable designing adaptive agents that change their roles or their behaviour in runtime; cooperative attitude of agents; and predefined architecture of the system. The paper ends in proposing some research axis such as finding new mechanisms, developing methods to design self-organising systems, providing means to control the global behaviour of the system, or proving convergence.

8 Acknowledgement

This work is partly supported by the Swiss NSF grant 200020-105476/1 and Agentlink III.

References

- [1] S. Ali R. Zimmer and C. Elstob. The question concerning emergence: implication for artificiality. In D.M. Dubois (Ed.), *First Computing Anticipatory Systems Conference (CASYS'97)*, CHAOS, Liège, Belgium, 1998.
- [2] O. Babaoglu, H. Meling and A. Montresor. Anthill: a framework for the development of agent-based peer-to-peer systems. In *Proceedings of the 22th International Conference on Distributed Computing Systems (ICDCS '02)*, pp. 15-22, IEEE Computer Society, Los Alamitos, CA, USA, 2002.
- [3] Y. Bar-Yam. *Dynamics of Complex Systems*. Perseus Books, Cambridge, MA, 1997.
- [4] E. R. Berlekamp, J. H. Conway, and R. K. Guy. *Winning Ways for your Mathematical Plays*, Volume 2, 2nd edition. AK Peters, Ltd., Wellesley, MA, 2001.
- [5] C. Bernon, V. Chevrier, V. Hilaire, P. Marrow, Applications of self-organising multi-agent systems: an initial framework of comparison. *Informatica*, Ljubljana, Slovenia. In press, 2005.
- [6] C. Bernon, M. Cossentino, J. Pavon. An Overview of Current Trends in European AOSE Research. *Informatica*, Ljubljana, Slovenia. In press, 2005.
- [7] E. Bonabeau, M. Dorigo, and G. Théraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Santa Fe Institute Studies on the Sciences of Complexity. Oxford University Press, New York, NY, USA, 1999.
- [8] L Bongaerts. *Integration of Scheduling and Control in Holonic Manufacturing Systems*, PhD Thesis, Katholieke Universiteit Leuven, 1998
- [9] R. Bordini et al. A survey on languages and platforms for MAS implementation. *Informatica*, Ljubljana, Slovenia. In press, 2005.
- [10] C. Bourjot, V. Chevrier, and V. Thomas. A new swarm mechanism based on social spiders colonies: from web weaving to region detection. *Web Intelligence and Agent Systems*, 1(1): 47-64, IOS Press, Amsterdam, The Netherlands, 2003.
- [11] S. Brueckner and H.V. Parunak. Self-organising MANET management. *Engineering Self-Organising Applications Systems*. G. Di Marzo Serugendo et al. (Eds), Lecture Notes in Artificial Intelligence, volume 2977, pp. 20-35. Springer-Verlag, Berlin, 2004.
- [12] S. Camazine, J.-L. Deneubourg, R. F. Nigél, J. Sneyd, G. Téraulaz, and E. Bonabeau. *Self-Organisation in Biological System*. Princeton Studies in Complexity. Princeton University Press, Princeton, NJ, USA, 2001.
- [13] D. Capera, J-P. Georgé, M-P. Gleizes, P. Glize. Emergence of organisations, emergence of functions. In *AISB'03 symposium on Adaptive Agents and Multi-Agent Systems*, pp 103 – 108, D. Kudenko, D. Kazakov, and E. Alonso (Eds), University of Wales, Aberystwyth, 2003
- [14] D. Capera, J. P. Georgé, M.-P. Gleizes, and P. Glize. The AMAS theory for complex problem solving based on self-organising cooperative agents. *International Workshop on Theory and Practice of Open Computational Systems (TAPOCS)*. Twelfth International IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE-2003), pp. 383-388. IEEE Computer Society Press, Los Alamitos, CA, 2003.
- [15] N. S. Contractor and D. R. Seibold. Theoretical frameworks for the study of structuring processes in group decision support system - adaptive structuration theory and self-organising systems theory. *Human Communication Research*, 19(4):528-563, 1993.
- [16] J-L., Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, L. Chrétien. The dynamics of collective sorting robot-like ants and ant-like robots - Simulation of animal behaviour. *Proceedings of the first international conference on simulation of adaptive behaviour*. J.A. Meyer and S. Wilson (Eds), pp. 356-363, MIT Press, Cambridge, MA, USA, 1991.
- [17] T. De Wolf and T. Holvoet. Adaptive behaviour based on evolving thresholds with feedback. *Proceedings of the AISB'03 Symposium on Adaptive Agents and Multiagent Systems*, pp. 91-96, D. Kudenko, D. Kazakov, and E. Alonso (Eds.), University of Wales, Aberystwyth, 2003.
- [18] T. De Wolf and T. Holvoet. Emergence and self-organisation: a statement of similarities and differences. *Engineering Self-Organising Systems*. S. Brueckner et al. (Eds), Lecture Notes in Artificial Intelligence, volume 3464, pp. 1-15, Springer-Verlag, Berlin, 2005.
- [19] G. Di Marzo Serugendo et al. Self-organising applications: paradigms and applications. *Engineering Self-Organising Systems*. G. Di Marzo Serugendo et al. (Eds), Lecture Notes in Artificial Intelligence, volume 2977, pp. 1-19. Springer-Verlag, Berlin, 2004.
- [20] G. Di Marzo Serugendo, A. Karageorgos, O. F. Rana, and F. Zambonelli (Eds). *Engineering Self-Organising Systems*. Lecture Notes in Artificial

- Intelligence, volume 2977, Springer-Verlag, Berlin, 2004.
- [21] G. Di Marzo Serugendo, M.-P. Gleizes, and A. Karageorgos. *AgentLink First Technical Forum Group Self-Organisation in Multi-Agent Systems*, AgentLink Newsletter, Issue 16, ISSN 1465-3842, pp. 23-24, 2004.
- [22] J. Dowling and V. Cahill. The K-Component Architecture Meta-Model for Self-Adaptive Software. *Proceedings of Reflection 2001*. Lecture Notes in Computer Science, volume 2192, pp. 81-88. Springer-Verlag, Berlin, 2001.
- [23] B. Edmonds. Using the experimental method to produce reliable self-organised systems. In *Engineering Self-Organising Systems*. S. Brueckner et al. (Eds), Lecture Notes in Artificial Intelligence, volume 3464, pp. 84-99. Springer-Verlag, Berlin, 2005.
- [24] S. Forrest. Emergent computation: self-organising, collective, and cooperative phenomena in natural and artificial computing network. *Proceedings of the Ninth annual CLNS conference*, 1990.
- [25] N. Foukia. IDReAM: Intrusion Detection and Response executed with Agent Mobility. *The International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'05)*, pp 264-270, Utrecht, The Netherlands, 2005.
- [26] P. Glansdorff and I. Prigogine. *Thermodynamic study of Structure, Stability and Fluctuations*. Wiley, 1971.
- [27] M.-P. Gleizes, V. Camps, and P. Glize. A theory of emergent computation based on cooperative self-organisation for adaptive artificial systems. *Fourth European Congress of Systems Science*. Valencia, 1999.
- [28] J. Goldstein. Emergence as a construct: history and issues. *Emergence* 1(1):49-72, ISCE Publishing, Mansfield, MA, USA 1999.
- [29] P. Grassé. La reconstruction du nid et les interactions inter-individuelles chez les bellicositermes natalenis et cubitermes sp. La théorie de la stigmergie: essai d'interprétation des termites constructeurs. *Insectes Sociaux*, 6:41-83, 1959.
- [30] S. Hassas, G. Di Marzo Serugendo, A. Karageorgos, C. Castelfranchi. Self-organising mechanisms from social and business/economics approaches, *Informatica*, Ljubljana, Slovenia. In press, 2005.
- [31] J.H. Holland. *Emergence – from order to chaos*. Addison-Wesley, Boston, MA, 1997.
- [32] O. Holland and C. Melhuus. Stigmergy, self-organisation, and sorting in collective robotics. *Artificial Life*, 5(2):173-202. MIT Press, Cambridge, MA, USA, 1999.
- [33] T. Ishida, L. Gasser, and M. Yoko. Organisation self-design in distributed production systems. *IEEE Transactions on Knowledge and Data Engineering*, 4(2): 123-134, IEEE Computer Society, Los Alamitos, CA, USA, 1992.
- [34] P. Jiang and Q. Mair. A self-organisational management network based adaptive resonance theory. *Agent Technologies, Infrastructures, Tools, and Applications for e-Services*. Kowalczyk et al. (Eds), *Lecture Notes in Artificial Intelligence*, volume 2592, pp. 211-225. Springer-Verlag, Berlin, 2003.
- [35] H. Karuna P. Valckenaers, B. Saint-Germain, P. Verstraete, C. B. Zamfirescu, H. Van Brussels, Emergent Forecasting using a stigmergy approach in manufacturing coordination and control. *Engineering Self-Organising Systems*. S. Brueckner et al. (Eds), Lecture Notes in Artificial Intelligence, volume 3464, pp. 210-226, Springer-Verlag, Berlin, 2005.
- [36] S. A. Kauffman, S.A. and S. Johnsen. Coevolution of the edge of chaos: coupled fitness landscapes, poised states, and coevolutionary avalanches. C.G. Langton et al. (eds.), *Artificial Life II*, Proceedings Volume X in the Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, Reading, MA, 1992.
- [37] A. Koestler. *The Ghost in the Machine*, Reprint edition, Penguin, East Rutherford, NJ, USA, 1990.
- [38] J. Liu, X. Jin, and K. C. Tsui. Autonomy oriented computing (AOC): formulating computational systems with autonomous components. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, In press, IEEE Computer Society, Los Alamitos, CA, USA, 2005.
- [39] P. Maes. Modeling Adaptive Autonomous Agents. C.G. Langton et al. (Eds), *Artificial Life 1(1-2):135-162*, MIT Press, Cambridge, MA, 1994.
- [40] M. Mamei, M. Vasirani, and F. Zambonelli. Self-organising spatial shapes in mobile particles: the TOTA approach. *Engineering Self-Organising System*. S. Brueckner et al. (Eds), Lecture Notes in Artificial Intelligence, volume 3464, pp. 138-153. Springer-Verlag, Berlin, 2005.
- [41] J-P. Mano, C. Bourjot, G. Leopardo, P. Glize. Bio-inspired mechanisms for artificial self-organised systems. *Informatica*, Ljubljana, Slovenia. In press, 2005.
- [42] F. Maturana and D. H. Norrie. Multi-agent mediator architecture for distributed manufacturing. *Journal of Intelligent Manufacturing*, 7:257-270. Kluwer Academic Publishers, Amsterdam, The Netherlands. 1996.
- [43] A. Montresor, H. Meling and O. Babaoglu. Messor: load-balancing through a swarm of autonomous agents. G. Moro and M. Koubarakis (Eds.) *Agents and Peer-to-Peer Computing*, Lecture Notes in Artificial Intelligence, volume 2530, pp. 125-137. Springer-Verlag, Berlin 2003.
- [44] J.-P. Müller. Emergence of Collective Behaviour and Problem Solving. *Engineering Societies in the Agents World - 4th International Workshop (ESAW 2003)*, A. Omicini, P. Petta, and J. Pitt (Eds), Lecture Notes in Artificial Intelligence, volume 3071, pp. 1-20. Springer-Verlag, Berlin, 2004.

- [45] H. V. Parunak and R. S. Vanderbok. Managing emergent behaviour in distributed control systems. *Proceedings of ISA Tech '97*, Instrument Society of America, 1997.
- [46] H. V. Parunak, J. Sauter, and S. Clark. Toward the specification and design of industrial synthetic ecosystems, *Intelligent Agents IV: Agent Theories, Architectures, and Languages*, Lecture Notes in Artificial Intelligence, volume 1365, pp. 45-59. Springer-Verlag, Berlin, 1998.
- [47] H. V. Parunak and S. Brueckner. Entropy and Self-Organisation in Multi-Agent Systems. In *International Conference on Autonomous Agents (Agents'01)*, 124-130. ACM Press, New York, NY, USA, 2001.
- [48] H. V. Parunak, S. Brueckner, J.A. Sauter: Digital pheromone mechanisms for coordination of unmanned vehicles. *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'02)*, pp. 449-450, ACM Press, New York, NY, USA, 2002.
- [49] H. E. Pattison, D. D. Corkill, and V. R. Lesser. Instantiating descriptions of organisational structures. *Distributed Artificial Intelligence*, pp. 59-96. M. N. Huhns (Ed.) Pitman, London, 1987
- [50] G. Picard, M-P. Gleizes. An agent architecture to design self-organising collectives: principles and application. *Proceedings of the AISB'02 Symposium on Adaptive Agents and Multi-Agent Systems*, Lecture Notes in Artificial Intelligence, volume 2636, pp. 141-158. Springer-Verlag, Berlin, 2002.
- [51] G. Picard, M-P. Gleizes. The ADELFE methodology - designing adaptive cooperative multi-agent systems. F. Bergenti, M-P. Gleizes, and F. Zambonelli (Eds). *Methodologies and Software Engineering for Agent Systems. The Agent-Oriented Software Engineering Handbook*, pp. 57-176, Kluwer Publishing, Amsterdam, The Netherlands, 2004.
- [52] G. Picard, C. Bernon, M-P. Gleizes. ETTO: emergent timetabling by cooperative self-organisation. *Proceedings of the third International Workshop on Engineering Self-Organising Applications (ESOA'05)*, pp. 31-45. Utrecht, The Netherlands, 2005
- [53] G. Poulton, Y. Guo, G. James, P. Valencia, V. Gerasimov, and J. Li. Directed self-assembly of 2-dimensional mesoblocks using top-down/bottom-up design. *Engineering Self-Organising Systems*. S. Brueckner et al. (Eds), Lecture Notes in Artificial Intelligence, Volume 3464, pp. 154-166. Springer-Verlag, Berlin, 2005.
- [54] R. Razavi, J. F. Perrot, and N. Guelfi. Adaptive modeling: an approach and a method for implementing adaptive agents. *Massively Multi-Agent Systems I*. T. Ishida, L. Gasser, H. Nakashima (Eds). Lecture Notes in Artificial Intelligence, volume 3446, pp. 136-148. Springer-Verlag, Berlin, 2005.
- [55] A. Reitbauer, A. Battino, A. Karageorgos, N. Mehandjiev, P. Valckenaers, and B. Saint-Germain. The MaBE middleware: extending multi-agent systems to enable open business collaboration. In *6th IFIP International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services (BASYS'04)*, 2004.
- [56] M. Schillo, B. Fley, M. Florian, F. Hillebrandt, and D. Hinck. Self-Organisation in Multiagent Systems: From Agent Interaction to Agent Organisation. *Third International Workshop on Modelling Artificial Societies and Hybrid Organisations (MASHO)*, pp. 37-46, 2002.
- [57] J. R. Searle. *The rediscovery of the mind*. MIT Press, Cambridge, MA, USA, 1992.
- [58] D. Servat, J. Leonard, E. Perrier, and J. P. Treuil. The Rivage project: a new approach for simulating runoff dynamics. J.Feyen and K.Wiyo, (Eds.) *Modelling of transport processes in soils*, pp. 592-601. Wageningen Press, Leuven, 1999.
- [59] M. Ulieru, Emergence of holonic enterprises from multi-agent systems: a fuzzy evolutionary approach. *Frontiers in Artificial Intelligence and Applications - Soft Computing Agents*, 83: 187-215. IOS Press - Frontiers in AI and Applications Series, Amsterdam, The Netherlands, 2002.
- [60] M. Ulieru. Adaptive Information Infrastructures for the e-Society. *Engineering Self-Organising Systems* S. Brueckner et al. (Eds), Lecture Notes in Artificial Intelligence, volume 3464 pp. 32-51. Springer-Verlag, Berlin, 2005.
- [61] F. Varela. *Principles of Biological Autonomy*. Elsevier, New York, NY, USA, 1979.
- [62] D. Weyns, K. Schelfhout, T. Holvoet, and O. Glorieux. Role based model for adaptive Agents. *Fourth Symposium on Adaptive Agents and Multi-agent Systems at the AISB '04 Convention*, 2004.
- [63] D. Weyns, T. Holvoet. On the role of environments in multiagent systems. *Informatica*, Ljubljana, Slovenia. In press. 2005.
- [64] F. Zambonelli, M.-P. Gleizes, M. Mamei, and R. Tolksdorf. Spray computers: frontiers of self-organisation for pervasive computing. *Second International Workshop on Theory and Practice of Open Computational Systems (TAPOCS 2004) in 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'04)*, pp. 397-402. IEEE Computer Society, Los Alamitos, CA, USA 2004.