SELF-ORGANISING APPLICATIONS: A SURVEY

Soraya Kouadri Mostéfaoui[φΨ], Omer.F Rana, Noria Foukia, Salima Hassas, Giovanna Di Marzo.S, Chris Van Aart, Anthony Karageorgos.

Engineering Self-Organising Applications Working Group

http://gaper.swi.psy.uva.nl/esoa

ABSTRACT

A self-organising system is a system that changes its basic structure as a function of its experience and environment. This definition relates to approaches undertaken in multi-agent systems, adaptive control, collective robotics, neural networks, and Grid computing research. The aim of this paper is to survey applications exhibiting emergent behaviour or complex social organisation, and outline the mechanisms enabling such behaviours.

Categories and Subject Descriptors

A.1 [**Introductory and Survey**], I.2.11 [**Distributed Artificial Intelligence**].

General Terms

Documentation, Design, Experimentation, Human Factors.

Keywords

Self-Organization, Emergence, Collective Behavior, Multi-Agent Systems.

1. INTRODUCTION

The essence of self-organisation is that the system structure (at least in part) appears without explicit pressure or constraints from outside the system. In other words, the constraints on form are internal to the system and result from the interactions between the components of a system. The organisation can evolve either in time or space, can maintain a stable form or can show transient phenomena. Thus it is reasonable to further demand that for a system to exhibit self-organising behaviour, its order cannot be imposed by special initial conditions, which would amount to a special creation. Self-organising behaviour is the spontaneous formation of well-organised structures, patterns or behaviours, from random initial conditions. Considerable research has already been undertaken to study such systems. Biology, chemistry, geology and sociology are some areas where self-organising systems are encountered often [1]. There is also a great deal of interest in information processing, knowledge based systems and Distributed Artificial Intelligence systems (DAI). Efforts to create networks of problem solving entities, utilising infrastructure such as the Grid, and the recent Agentcities project, lead to the problem of steering, maintaining and coordinating large communities of heterogeneous software agents. This survey first gives the basic notions of self-organisations. Second it describes the different mechanisms enabling social organisations to achieve a coherent global behaviour through local interactions. Third, it reviews several applications exhibiting a self-organising behaviour. Focus is given on applications belonging to the following domains: artificial life, robots, networking applications, Grid computing, and the Agentcities network.

2. BACKGROUND

This section explains the basics of self-organizations through an illustrative example, and describes the main characteristics of self-organizing systems.

2.1 Magnetization

To make the phenomenon of self-organisation more concrete, it is useful to look at basic examples that are the inspiring models for today's self-organising applications. Perhaps the simplest such process that has been extensively studied is magnetization[1]. A piece of potentially magnetic material consists of a multitude of tiny magnets, called "spins". In general, these spins will point in different directions, so that their magnetic fields cancel each other out. The random movements of the molecules in the material cause this disordered configuration. The higher the temperature, the stronger these random movements affecting the spins, and the more difficult it will be for any ordered arrangement of spins to maintain or emerge. However, when the temperature decreases, the spins will spontaneously align themselves, so that they all point in the same direction (Figure 1). Instead of canceling each other, the different magnetic fields now add up (reinforce), producing a strong overall field. Magnetization is a clear case of self-organisation, which can be used as a paradigm for a whole range of similar phenomena, such as crystallization where not only the orientations, but also the positions of the molecules become evenly arranged.

2.2 Characteristics of Self-Organising Systems

Self-organisation has three important characteristics. First, a self-organising system can accomplish complex tasks by integrating simple individual behaviours of its constituents. Secondly, a change in the environment may influence the same system to generate a different task, without any change in the behavioural

---

[φ] To whom correspondence should be addressed.

[Ψ] Soraya.Kouadri.M is with the Parallelism and Artificial Intelligence Research group (PAI) at the university of Fribourg Switzerland. Soraya.kouadrimostefaoui@unifr.ch

[1] http://pespmc1.vub.ac.be/SELFOREX.html

characteristics of its constituents. Finally any small differences in individual behaviour of constituents can influence the collective behaviour of the system [2].

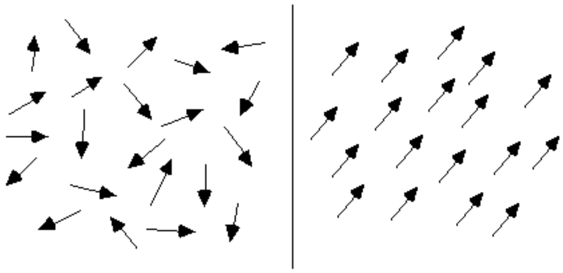

Figure 1. Two arrangements of spins: disordered (left) and ordered (right).

2.3 How Can Self-Organisation be Studied?

Since we are seeking general properties that apply to topologically equivalent systems, we start in general with a set of rules specifying how the interconnections are allowed to behave. The network is randomly initiated and then iterated, the stable pattern observed is noted and the sequence repeated. After many trials (or iterations), generalisations from the results can be derived, with some statistical probability.

3. SELF-ORGANISATION MECHANISMS

This section presents the major self-organising mechanisms used in nowadays applications.

3.1 Self Organising Maps (Kohonen Networks)

Self-organisation of neuronal functions seems to exist on very abstract levels in the brain. When a laboratory rat has learned its location in a labyrinth, certain brain cells on the hippocampal cortex respond only when it is in a particular location. The Kohonen Self-Organizing Map (SOM) (Kohonen 1997) has a similar principle: units (referred to as neurons) are recruited topologically for tasks depending on the sensory input. It is commonly classified as a neural network, and more specifically a winner-takes-all competitive algorithm, since the units *compete* with each other for specific tasks. Each unit i has its own prototype vector w_i (also referred to as codebook vector or weight vector), being a local storage for one particular kind of input vector that has been introduced to the system. Initially these prototype vectors with a dimension n equal to the input space start out as vectors with random small components and, as new input enters the SOM, are improved following this update rule:

$$w_i = w_i + \alpha \cdot \eta(winner) \cdot (x_i - w_i), \forall i \in \{1,...,n\}$$

Where α is called the learning rate and lies between 0 and 1, and η (*winner*) is the neighborhood function ranging from 0 to 1 as well, depending on the distance between the current SOM unit and the winner. The winner is the unit that has a prototype vector that is closest to the current input vector using the Euclidean distance:

$$winner = \arg\min_j \sqrt{\sum_{j=1}^{n} (x_j - w_j)^2}$$

The neighborhood function is traditionally implemented as a Gaussian (bell-shaped) function:

$$\eta(winner) = \frac{1}{\sqrt{2\pi}nb} e^{-0.5 \cdot (winner - current)^2 / nb^2}$$

Where *nb* is a parameter indicating the width of the function, and thus the radius in which the neighbors of the winning units are allowed to update their prototype vectors significantly [3]. The map of units is usually taken as a two dimensional grid, although many other organisations have been applied (such as a map of hexagons). After a sufficient amount of input data has been presented to the SOM, self-organization will result in a topographic map, where similar data is mapped onto units in a particular region of the map, and neighboring units will be activated (i.e. become winners) for similar input data. Figure 2 shows how different units become recruited for different states of the environment by colouring the units according to the state in which they were declared as winners.

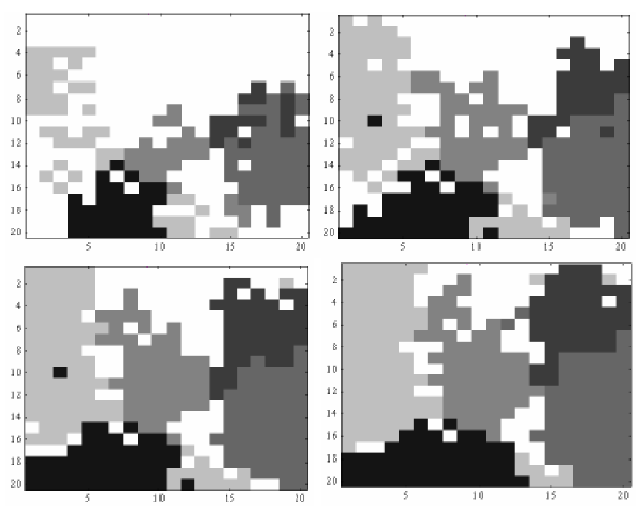

Figure 2. A 2D Self-Organizing Map showing different regions.

3.2 Social Insects Paradigm

Social insect societies (ants, bees, wasps, termites, etc) exhibit many interesting complex behaviors, as emergent proprieties from local interactions between elementary behaviours achieved at an individual level. The emergent collective behavior is the outcome of a process of self-organization, in which insects are engaged through their repeated actions and interactions with their evolving environment [4]. Self-organization in social insects relies on an underlying mechanism, the mechanism of stigmergy, first introduced by Grassé in 1959 [5]. Grassé studied the behavior of a kind of termites during the construction of their nests and noticed that the behavior of workers during the construction process is influenced by the structure of the constructions themselves. This mechanism is a powerful principle of cooperation in insect societies. It has been observed within many insect societies like those of wasps, bees and ants. It is based on the use of the environment as a medium of inscription of past behaviors effects, to influence the future ones. This mechanism defines what is called a self-catalytic process, that is the more a process occurs, the more it has chance to occur in the future. More generally, this mechanism shows how simple systems can produce a wide range of more complex coordinated behaviors, simply by exploiting the influence of the environment. Many behaviours in social insects, such as foraging or collective clustering are rooted on the stigmergy mechanism. Foraging is the collective behavior through which ants collect food by exploring their environment. It is based on the stigmergy mechanism. During the foraging process, ants leave their nest and explore their environment following a random path. When an ant finds a source of food, it carries a piece of it and returns back to the nest, by laying a trail of a hormone called *pheromone* along its route. This chemical substance persists in the environment for a particular amount of time before it evaporates. When other ants encounter a trail of pheromone, while exploring the environment, they are influenced to follow the trail until the food source, and enforce in their coming back to the nest the initial trail by depositing additional amounts of pheromone. The more a trail is followed, the more it is enforced and has a chance of being followed by other ants in the future. Collective Sorting is a collective behavior through which some social insects sort eggs, larvae and cocoons [6]. As mentioned in [7], an ordering phenomenon is observed in some species of ants when bodies of dead ants are spread in the foraging environment. Ants pick up dead bodies and drop them later, in some area. The probability of picking up/depositing an item is correlated with the density of items in the region where the operation occurs. This behavior has been studied in Robotics through simulations [8] and real implementations [4]. Robots with primitive behavior are able to achieve a spatial environment structuring, by forming clusters of similar objects via the mechanism of stigmergy described above. Social insects provide a new paradigm for developing decentralized complex applications such as autonomous and collective robotics, computing systems for networks and telecommunications, optimization algorithms, etc.

3.3 Coordination in Multi-Agents Systems

After defining what Multi-Agents Systems (MAS) are, this subsection lists some mechanisms, used for MAS and robots coordination that enables a group of agents or robots to exhibit a self-organising behavior.

3.3.1 Multi-Agent Systems

Distributed Artificial Intelligence (DAI) has looked at overcoming limitations of individual agencies by tacking problems through running distributed computational processes. DAI is a sub-field of Artificial Intelligence and has for more than two decades been investigating concurrency within AI computation, at many levels [25]. Based on this research in (distributed) knowledge models, communication and reasoning techniques have led to ways for agents to participate in societies of agents, i.e. agencies. Examples of agencies are collection of individuals, including humans, machines and computational processes such as web services and agents. We see an agency as a society of agents, in which each of them can be specialized with knowledge, one or more skills and has a sort of mechanism with which it can interact with others. A specific agency is a multi-agent system, which is defined by O'Hare and Jennings as a loosely coupled network of problem solvers that work together to solve problems that are beyond their individual capabilities [26]. Due to its emergent behaviour, a multi-agent system naturally exhibits self-organising characteristics.

Every agent has one or more limitations, which can be categorized into cognitive limitations, physical limitations, temporal limitations and institutional limitations. Cognitive limitations resemble the fact that individuals are rationally bounded. It means that the data, information, and knowledge an individual can process and the detail of control an individual can handle is limited. As tasks grow larger and more complex, techniques must be applied to limit the increase of information and the complexity of control. Individuals can be limited physically, because of their physiology or because of the resources available to them. Temporal limitations exist where the achievement of individual goals exceeds the lifetime of an individual, or the time over which resources are available for achieving a goal. Finally, individuals can be legally or politically limited.

3.3.2 Coordination Models

A coordination model is a formal framework useful to study and understand problems in designing programming languages and software architectures comprising several problem solvers (agents). In other words, a coordination model defines *how agents interact and how their interactions can be controlled* [10]. This includes dynamic creation and destruction of agents, control of communication flows among agents, control of spatial distribution and mobility of agents, as well as synchronization and distribution of actions over time [11]. In general a *coordination model* is defined by a triple (E, M, L), where:

- E are the *coordinable entities* (components): these are the agents, which are co-coordinated. Ideally, these are the building blocks of a co-ordination architecture (e.g. agents, processes, tuples, atoms, etc.)
- M are the *coordinating media (connectors)*: these are the co-coordinators of inter-agent entities. They also serve to aggregate a set of agents to form a *configuration*. (e.g. channels, shared variables, tuple spaces)
- L are the *coordination laws* ruling actions by co-coordinable entities or the coordination media. Usually the laws define the semantics of a number of co-ordination mechanisms that can be added to a host language.

Co-ordination models differ mostly in the way they control interaction: for instance, different models could offer different kinds of mobility:

- ***Planned***: an agent's itinerary across some locations is statically predefined;
- ***Spontaneous****:* an agent's itinerary is not statically predefined, but the next location is computed by the agent itself at runtime;
- ***Controllable****:* a migration is forced by an authority in some location, using some I/O mechanisms to communicate with a

remote agent. Interestingly, there are two types of controllable mobility: *sender-controlled* and *receiver controlled*.

Usually a coordination language has to be combined with a conventional programming language to obtain a fully-fledged programming language. A number of coordination languages have been defined and studied in the last decade; however the field is far from being exhausted, especially because the concept of "coordinable entity"", or agent, has still to be fully understood, and increases in complexity of infrastructure over which coordination is achieved.

3.3.3 Some dimensions of coordination.

The basic ideas in all coordination models and languages are: "*minimalism*" (a small set of coordination primitives should suffice) and "*optimizability*" (it should be possible to reason on and compile co-ordination primitives) [11]. Coordination models and related languages can be classified along a number of dimensions:

- Location-less vs locality-based (named) coordination media
- Transactional (multi-set based) vs asynchronous (tuple based)
- Coordination media.
- Procedural (imperative, functional, or logic) vs object-oriented (or agent-oriented) coordinables.
- Centralized vs decentralized co-ordination laws.
- Data-driven vs event-driven co-ordination primitives.

4. Self-Organising Applications

Many studies of complex systems assume that the systems self-organise into emergent states which are not predictable from the individual parts. Artificial life, collective robotics, evolutionary computing, cellular automata, and neural networks are the main fields directly associated with this idea, and in which a large number of applications has been done. In this section we take a look to some systems that exhibit a Self-Organising behavior.

4.1 Grid

Computational Grids provide the software and networking infrastructure required to integrate computational engines/scientific instruments, data repositories, and human expertise to solve a single large problem (generally in science and engineering domains). Computational engines can comprise of specialist, tightly coupled architectures (such as parallel machines) or loosely coupled clusters of workstations. There has been an emerging interest in trying to integrate resources across organizational boundaries through file or CPU sharing software (such as KaZaA and Gnutella for file sharing and Entropia and UD for CPU sharing). Often these individual resources are geographically distributed, and may be owned by different administrators (or exist within different independently administered domains).

Managing resources within Computational Grids is currently based on infrastructure with centralized registry and information services (based on the LDAP/X500 directory service) – such as provided by the Open Grid Services Infrastructure (OGSI). In this process, resource owners must register their capabilities with a limited number of index servers, enabling subsequent search on these servers by resource users. The provision of such centralised servers is clearly very limiting, and restricts the scalability of such approaches. Although current resources being provided within Computational Grids are owned by national or regional centers (or by research institutions), and therefore concerns regarding access rights and usage need to be pre-defined and approved. However, as resources from less trusted users are provided, the need to organize these into dynamic communities, based on a number of different criteria: performance, trust, cost of ownership and usage, usability etc become significant. Self- organisation therefore plays an important role in identifying how such communities may be formed, and subsequently dis-banded. A utility-based approach for forming such communities is explored in [12]. However, it is necessary to understand and investigate alternative incentive structures that will enable the formation of such communities. Recent interest by IBM, as part of their "Autonomic Computing" program, and by Microsoft, as part of the "Dynamic Systems Initiative", indicates the importance of self-organisation for managing distributed resources.

4.2 Networking Applications

This subsection presents a range of self-organizing applications related to the networking domain.

4.2.1 *Adaptive Networking and Service Emergence*

Itao et al. [13] envision a future where a universal network connects every human being and most human-made electronic devices. The universal network is supposed to span locations engaged in every human endeavor, including the home, workplace, transportation vehicles, public facilities, and space facilities. To realize this vision, they propose a radically new paradigm of the adaptive networking architecture for service emergence called Jack-in-the-Net (Ja-Net). The Architecture is inspired by the observation that the biological world has already developed the mechanisms necessary to achieve such key requirements as self-organisation, scalability, adaptation and evolution, security, and survivability necessary for the envisioned universal network. In the proposed architecture, network services and applications were implemented by a distributed, adaptive, and self-organising collective entity called the super-entity, which consists of a large number of autonomous entities called cyber-entities (analogous to a bee colony consisting of multiple bees). Each cyber-entity implements a functional component related to the overall service or application and follows simple behaviour rules (e.g., migration, replication, reproduction, pheromone emission, energy exchange, mutation, death) similar to biological entities. Useful emergent behaviours (e.g., scalability, adaptation, evolution, security, survivability, and simplicity) result when individual cyber-entities interact [14].

4.2.2 Self-Organisation and Identification of Web Communities

G. William Flake et al. [15] have recently shown that despite the decentralized, unorganized, and heterogeneous nature, the web self-organizes such that the link structure allows efficient identification of communities. They modeled the web as a graph where vertices are the web pages and hyperlinks are the edges.

They define a web community as a collection of web pages such that each member page has more hyperlinks (in either direction) within the community than outside of the community. This definition was generalised to identify communities with varying sizes and levels of cohesiveness. Community membership is defined as a function of both a web page's outbound hyperlinks and other hyperlinks on the web; therefore these communities are "natural" in the sense that they are collectively organised by independently authored pages. G. William Flake etal. noted that using only link information to identify a naturally formed community -according to the definition- is intractable in the general case because the task maps into a family of NP-complete graph partitioning problems. To deal with that the authors assumed the existence of one or more seed web sites and have exploited systematic regularities of the web graph, thus allowing the problem to be recast into a maximum flow framework that allows an efficient community identification using a polynomial time algorithm that scale well to studying the entire web graph (Figure. 2).

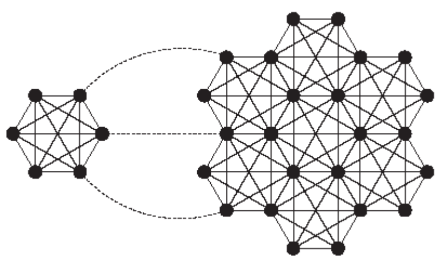

Figure 2. A simple community identification example. Maximum flow methods will separate the two sub graphs with any choice of source vertex *s* from left and sink vertex *t* from the right sub graph, removing the three dashed links. As formulated with standard flow approaches, all community members must have at least 50% of their links inside of the community; however, additional artificial links can be used to change the threshold from 50% to any other desired threshold. Thus, communities of various sizes and with varying levels of cohesiveness can be identified and studied.

4.2.3 Networking with Ants

Dorigo suggests to use artificial ants modeling to solve network problems [16]. The motivation is that ants modeling might be able to cope with communication networks better than humans. A first survey, dealing with several swarm intelligence examples in social insect communities, shows how ants-like behaviour (ants, bees, termites, and wasps) provide a powerful metaphor to build a completely decentralized system. Such a system is composed of individual and simple entities, which collaborate to allow a more complex and collective behaviour [17]. The global emergent behaviour of the ant population is due to a network of interactions between the ants themselves but also between the ants and their environment. This emergent collective behaviour allows the social insect colony to organize vital task like finding food, building the nest, dividing labor among individual, spreading alarm, among all. Many of those tasks and their respective mechanism have inspired computer (network) scientists notably to mimic ant foraging behaviour to optimize the routing in communication networks or to mimic the division of labor and the task allocation to optimize the load balancing in network systems. When ants forage, they first randomly wander the floor from the source (the nest) to the destination (the food) and they deposit/lay a chemical trail (the so-called pheromone. The pheromone deposited along the path followed by each ant marks this path for other ants. In fact, the more one path is marked the more it will be chosen by other ants. This mechanism where the environment becomes the communication medium is called stigmergy. To apply this paradigm to network routing Dorigo and his colleagues built an artificial ant network where periodically each node launches an ant to find the route to a given destination. By simply smelling the strength of the pheromones along the neighborhood paths of the node, the ant generates the map that shows the fastest route to any end point. In case of congestion, it was showed that this mechanism outperform all other popular routing models/systems/algorithm in term of speed, to avoid the traffic jams.

4.2.4 Network Security (Mobile Agents for Intrusion Detection and Response)

The use of Mobile Agents (MAs) in sophisticated applications could offer advantage for constructing flexible and adaptable wide-area distributed systems. Notably, applications such as Intrusion Detection Systems (IDSs) and Intrusion Response Systems (IRSs) have become even more relevant in the context of large-scale network infrastructures, where traditional security mechanisms demonstrate severe weaknesses [18]. Indeed, as they can be retracted, dispatched, cloned or put in stand-by, MAs have the ability to sense network conditions, and to load dynamically new functionalities into a remote network node (such as a router). The network awareness of MAs can also significantly contribute to the detection of, and enable response to intrusions. Therefore we propose MA technology as support for Intrusion Detection (ID) and Intrusion Response (IR) in computer networks, which is still a relatively unexplored area. We concentrate on aspects of ID and IR where the self-organizing properties of MAS are of particular benefit. Indeed, the originality of the adopted approach lies on the design of the IDRS, where the organisation of MAs follows the behaviour of natural systems to detect an intrusion as well as to answer an intrusion as it is described in [20].

Schematically there are two natural paradigms that have been referred to:

- First, the human immune system because the IDS is based upon principles derived from the immune system model where Intrusion Detection Agents (ID Agents) map the functionalities of the natural immune system to distinguish between normal and abnormal events (respectively "self" and "non self" in the immune system) as explained in [19].

- Second, the social insect stigmergy paradigm because the IRS is based upon principles derived from this paradigm. In fact, Intrusion Response Agents (IR Agents) map the collective behaviour of an ant population by following a synthesized electronic pheromone specific to the detected intrusion until the source of the attack -- in order to perform its response task. This pheromone has been previously diffused throughout the network by an ID Agent when it detected the attack This kind of collective paradigm is very interesting because it consists in having each ant execute a rather light task (MAS play the role of ants in the IR System) to induce collectively a more complex behaviour; this approach also is very powerful because the ID System as well as the IR System are completely distributed in the network, without any centralized control: both systems are essentially constituted by MAS which travel across the network, dynamically adjusting their routes according to collected events, without any simple way to trace them. Besides, our MAs are quite polyvalent because they can detect and/or respond to intrusion. This enhances the difficulty for an attacker to distinguish between ID Agents and IR Agents.

4.3 Robots

Modeling Army-ant robots as a self-organising system has many advantages. Adaptive, collective and "complex" systems resulting from simple individual behaviour are what the Army-ant scenario envisages. Simplicity of the individual agents is an important factor in implementation. However, the size and non-linear character of self-organisation leaves little possibility for definitive analysis. It is possible to geometrically arrange Army-ant robots by using a distributed approach. Robots can spatially organize themselves around a goal using only local information transferred by broadcast signals. This method is more advantageous and faster than conventional centralised control methods, especially when the number of agents is large. The methods described in the text can be applied directly to other multi-robot systems in underwater, planetary surface and space missions. It is also possible to separate the agents into different teams around different goals. The size of these teams can be determined by the difficulty of the assigned task. The team formation can again be achieved by using broadcast signals. Although there is no hierarchy between agents, temporary "*leader*" assignments seem to be necessary to overcome several problems. However, this is not a violation of the homogeneous character of the population since all agents may become one and replace the leader. Self-organising mobile robots need to be equipped with communication devices as well as beacons and detectors; the use of communication channels enables co-operating robots to form a decision mechanism. Army-ant robots can share individual information, and consequently "act" intelligent. Driven by several behavioral modules, Army-ant robots form a large dynamic system. Interaction "rules" between agents have to be adjusted or have to self-adjust carefully to the environment and/or tasks to define the "responsibilities" of agents during different phases[2].

4.4 Artificial Life

Lately much attention has been posited on evolutionary strategies that bring together self-organising systems and natural selection inspired algorithms. Particularly in the field of artificial life, [21] and [22] have proposed a genetic algorithm which does not encode directly their solutions, but rather encode genetic rules which develop into Boolean networks simulating given metabolic cycles. With these approaches, genetic algorithms no longer model exclusively selection, but also a self-organising dimension standing for some materiality.

The genetic algorithm does not search the very large space possible solutions, but a space of basic rules which can be manipulated to build different self-organising networks which will themselves converge to a solution --*emergent morphology* [23].

4.5 Emergent Manufacturing Control Using Ant Colony Techniques

The food foraging behaviour in ant colonies has been translated into a design for agent societies performing manufacturing control. Resource agents provide a reflection of the underlying production system in the world of the agents. These resource agents offer a space for the other agents to navigate through - each agent knowing its neighbors – and offer spaces on which information can be put, observed and modified - like the ants leave pheromones in the physical world.

Virtual agents - ants - move through this reflection of the physical world and collect information, which they make available elsewhere. First, these ants collect information about the available processes, travel upstream and place the collected information at routing points. Second, ants explore possible routings for the products being made, make a selection and propagate the resulting intentions through the 'reflection'. Resource agents receive this information about the intentions of their users and compile short-term forecasts for themselves. These forecasts allow up-to-date predictions of processing times used by the ants exploring routes and propagating intentions.

All these activities are subjected to an evaporation (time-out) and refresh process that enables the system to keep functioning in a highly dynamic environments (frequent changes and disturbances) [24].

4.6 The AgentCities Network

4.6.1 What is AgentCities?

Agentcities is a worldwide initiative designed to help realize the commercial and research potential of agent based applications by constructing a worldwide, open network of platforms hosting diverse agent based services. The ultimate aim is to enable the dynamic, intelligent and autonomous composition of services to achieve user and business goals, thereby creating compound services to address changing needs.

The initiative will build on a wealth of innovative technologies including agent technology, Semantic Web technologies, UDDI discovery services, eBusiness standards and Grid Computing. Application areas already envisaged range from eHealth and eLearning to manufacturing control, digital libraries, travel and entertainment services. The Agentcities network is designed to:

- Act as a distributed testbed for experimenting with Agent technology and composable services.
- Create a common resource for developers wishing to collaborate with each other and link up their agent systems and services.
- Provide a benchmark environment to validate and test compliance to relevant technology standards and provide input to the standards themselves.
- Act as a focus for discussion of next generation information networks as well as the development of services, technologies and methodologies.

---

[2] http://armyant.ee.vt.edu/unsalWWW/thesisbiblio.html

4.6.2 Application Scenario: A Self-Organised Document Retrieval System for the Agentcities Network

In this section we take a look to our initial efforts to apply a self-organizing mechanism to the Agentcities Network. This system is a portal that provides access to a set of documents (pdf, ps, doc…etc) ***integrated*** and ***organised***. It is based in topic categorization and supports retrieval of documents based on query words. It employs a Self-Organising Map (SOM) to automatically organize the documents in a theme map. The documents are clustered on the basis of the frequency distribution of the words used. Through the generated theme map the system provides a range of services such as retrieval of documents by query words, intuitive browsing of the documents by theme inspection, automatic generation of hypertext links to related sites…etc.

In summary the system is be able to do:

- An automatic clustering of documents into themes;
- Order these clusters in a theme map using Kohonen self-organising maps;
- Extract meaningful labels for each cluster of documents;
- Use the extracted labels to retrieve ranked lists of documents based on query words;
- Summarize the documents;
- Generate automatically links to related documents hosted in other related sites.

5. CONCLUSION

What makes a self-organising system advantageous over a pre-programmed, deterministic organisation is that the former is based on agents requiring simple programming and autocatalytic communications. A large number of individuals can be co-coordinated into a collective system interacting with an environment; this collective behaviour will have an adaptive character. Such a system is therefore simple, reliable and adaptive where only few basic rules are needed to define individual behaviour and interactions. Furthermore, breakdown of one agent will not affect the activity of the whole system, which may not be the case in deterministic systems. The simplicity would also extended to the software as well as hardware required to implement the system. In a deterministic system, programs are highly complex, as it is necessary to specify behaviours to respond to every possible situation that a system may encounter, and it is still impossible to foresee them all. However in a self-organising system, simpler programs can operate in unforeseen situations and adapt to changing conditions. For these reasons, self-organising algorithms which have only partial (local) knowledge of the network are used to manage data networks of large numbers of users.

Multi-agent systems (MAS) are collections of interacting autonomous entities. The behaviour of the MAS is a result of the repeated asynchronous action and interaction of the agents. Understanding how to engineer self-organisation is thus central to the application of agents on a large scale. Multi-agent simulations can also be used to study emergent behaviour in real systems.

Interest in large-scale systems of agents is growing, and advances in telecommunications and the spread of the Internet and electronic commerce means that information infrastructure is now required to operate as a global dynamic system. Furthermore, the density and diversity of interconnections in such system will increase rapidly over time. Individual system administrators are also not able to see optimisations of an entire system (as each component within the system gets more complex), and there is therefore a need to enable components themselves to support self-organisation. Moreover, such systems are being required to serve the needs of a diverse set of users (whatever their distinctive needs), not just a virtual "representative" user. Thus, such systems must adapt to personal requirements, by providing highly customized packages of services. Simultaneously providing highly diverse services to a huge user population in an enormous, interconnected system is a task beyond centralised management techniques.

A useful way to manage this form of agent-based system is to utilize its emergent properties to make it self-organising and self-regulating. Desirable self-organisation is observed in many biological, social and physical systems. However, fostering these conditions in artificial systems proves to be difficult and offers the potential for undesirable behaviours to emerge. Thus, it is vital to be able to understand and shape emergent behaviours in agent based systems. Current mathematical and empirical tools give only a partial insight into emergent behaviour in large, agent-based societies.

6. ACKNOWLEDGMENTS

The research described in this paper is partly supported by the EC project Agentcities.RTD (IST-2000-28385) the authors would like to thank this support. The opinions expressed in this paper are those of the authors and are not necessarily those of the EU Agentcities.RTD partners.

7. REFERENCES

[1] Giovanna, D.S and al. Self-Organizing Applications a Survey. Agentcities Task Force Technical Note, March 2003.

[2] Soraya. Kouadri.M. Collective Adaptation of a Heterogeneous Communicating Multi-Robot System. Proceedings of the 2002 International Arab Conference on Information Technology ACIT'02. University of Qatar, Doha, Qatar, December 2002. PP 1038-1044.

[3] E. Catterall, K. Van Laerhoven and M. Strohbach. "Self-Organization in Ad-Hoc Sensor Networks: An Empirical Study". In Proc. of Artificial Life VIII, The 8th International Conference on the Simulation and Synthesis of Living Systems. Sydney, Australia 2002. MIT Press.

[4] O. Holland, C. Melhuis, "Stigmergy, Self-Organization and Sorting in Collective Robotics", in Artificial Life (5), pp. 173-2002, MIT, 1999.

[5] P-P. Grassé, "La reconstruction du nid et des coordinations inter-individuelles chez Bellicositermes matalensis et Cubitermes sp. La théorie de la stigmérgie:: essai

d'interprétation des termites constructeurs.", in Insectes Sociaux, 6, pp. 41-83., 1959 (in French).

[6] J.-L., Deneubourg., S., Goss., N., Francks., A., Sendova-Francks., C., Detrain, & L.Chrétien., "The dynamic of collective sorting: Robot-like ants and ant-like robots". In Proc. of the First International Conference on the Simulation of Adaptive Behavior: From Animals to Animats (). Cambridge, MA, MIT Press, 1991, pp. 356-363.

[7] E. Bonabeau, V. Fourcassié, J.L. Deneubourg, The phase ordering Kinetics of cemetery organization in ants, Research Report, Sante Fe Institue, 1998.

[8] O., Holland, "Multi-agents systems: Lessons from social insects and collective robotics." In Adaptation, Co evolution and Learning in Multi-agents Systems: Papers from the 1996 AAAI Spring Symposium Tech.

[9] M. Dorigo, V. Maniezzo & A. Colorni, "The Ant System: Optimization by a Colony of Cooperating Agents.", IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1): pp 29-41, 1996.

[10] Paolo Ciancarini, Multiagent coordination: A Computer Science perspective, MAAMAW, Annecy, May 2001

[11] N. Carriero and D. Gelernter, "Tuple Analysis and Partial Evaluation Strategies in the Linda Precompiler," Languages and Compilers for Parallel Computing, MIT Press, 1990, pp. 115- 125.

[12] Coordinated Learning to support Resource Management in Computational Grids, Steven Lynden and Omer F. Rana, Proceedings of 2nd IEEE Conference on Peer-2-Peer Computing, Linkoping, Sweden, Sept. 2002.

[13] T. Itao, T. Suda, T. Nakamura, M. Imada, M. Matsuo Jack-in-the-Net: an Adaptive Networking Architecture *for Service Emergence.*. Proceedings of teh Asian-Pacific Conference on Communications, 2001.

[14] T. Suda. Larsge. Adaptive Networking Architecture for Service Emergence. Scale Networking Workshop. March 2001.

[15] G. William, S. Lawrence, C. Lee Giles and F. M. Coetzee. Self-Organisation and identification of web communities. In IEEE Computer, March 2002.

[16] M. Dorigo and G. Di Caro. Ants colonies for adaptive routing in packet-switched communication networks. Lecture Notes in Computer Science, page 673, 1998.

[17] E. Bonabeau, M. Dorigo and G. Theraulaz. Swarm Intelligence: From Natural to Artificial Systems. In Oxford University Press, Series: SFI Studies in the Sciences of Complexity, July 1999.

[18] S. Martino, A Mobile Agent Approach to Intrusion Detection, Joint Research Center-Institute for Systems, Informatics and Safety, Italy, June 1999

[19] S. Forrest, S. Hofmeyr, A. Somayaji, and T. Longstaff. A sense of self-Unix processes, 1996.

[20] N. Foukia, S. Hassas, S. Fenet, J. Hulaas: An Intrusion Response Scheme: Tracking the Source using the Stigmergy Paradigm, in proceedings of SECURITY OF MOBILE MULTIAGENT SYSTEMS WS (SEMAS-2002), Bologna, Italy, July 16 2002.

[21] R. A. Brooks and P.Maes. Evolution of metabolism for Morphologies. H. Kitano. In Artificial Life IVMIT Press, pp 49-58, 1994.

[22] F. Dellaert and R.D. Beer. Toward an evolvable model of development for autonomous agent synthesis. In Artificial Life IV. Brooks, R.A and P.Maes (Eds). MIT Press, 1994.

[23] L. M. Rocha. Selected Self-Organisation and the semiotics of Evolutionary Systems. In: Evolutionary Systems: The Biological and Epistemological Perspectives on Selection and Self- Organisation, . S. Salthe, G. Van de Vijver, and M. Delpos (eds.). Kluwer Academic Publishers, pp.341-358, 1998.

[24] P. Valckenaers, H. Van Brussel, M. Kollingbaum, and O. Bochmann Multi-agent Coordination and Control Using Stigmery Applied to Manufacturing Control.. In Multi-Agent Systems and Applications, 9th ECCAI Advanced Course ACAI 2001 and Agent Link's 3rd European Agent Systems Summer School, EASSS 2001, Selected Tutorial Papers. M. Luck, V. Maík, O. tpánková, R. Trappl (Eds.), LNAI 2086, pp.317-335. Prague, Czech Republic, July 2-13, 2001.

[25] Readings in Distributed Artificial Intelligence. A. Bond, and L. Gasser. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1988.

[26] Foundations of Distributed Artificial Intelligence. G. O'Hare, and N. Jennings. John Wiley and Sons, 1996.

[27] http://www.agentcities.org/About/index.jsp