A Platform for the analysis of artificial self-organized systems

Christine Bourjot, Vincent Chevrier
LORIA Campus Scientifique
BP 239
54506 Vandoeuvre-Les-Nancy CEDEX France
{bourjot, chevrier}@loria.fr

***Abstract*— The context of this work is that of systems being able to produce a collective response from interaction between simple individuals. Such systems are qualified of self organized ones and can be modeled with reactive multi-agent systems.**

This article underlines why, from our point of view, the analysis of these systems requires an experimental approach and it details our proposition in terms of tools for experimentations. After commenting each main components of the platform, it provides some details about its current implementation in the case of two systems.

I. Introduction

The context of this work is the study, the analysis and the design of systems being able to produce a collective response from interactions between simple individuals. What is interesting is the ability of such systems to have a collective response such as its complexity is out of the individual simplicity. Therefore this complexity is not in the individuals but in the interactions between them. Such systems can be qualified of self-organized.

Camazine et al [1] define self-organization as a process in which pattern at the global level emerges solely from numerous interactions among lower-lever components of the systems. Self-organized mechanisms are robust, decentralized, and can resist to perturbations.

These systems are characterized mainly by non linear dynamics (small fluctuations when near some critical point can provide significant modification of the system), by sensibility to initial conditions and parameter sensibility (small changes on a parameter produce different patterns). Thus the overall properties cannot be understood simply by examine separately the components.

For example, the following (and well known) equation models a population (which varies from 0 to 1) at time n: $x_{n+1} = Rx_n(1 - x_n)$ where R is the reproductive rate and is a parameter. In this example, bifurcation occurs at several values of R. The population size to which the system converges is the attractor. For a given value of R, several attractors can be exhibited depending on the range of the initial conditions.

Furthermore, formal studies of them are often difficult and are restricted to specific cases or are complementary [2] with other approaches such as multi-agent models. Multi-agent models [3] provide concepts to describe a system as a set of agents that act in a shared environment and which interact together and via their environment. Reactive multi-agent systems [4] are systems made of simple behaving units with decentralized control. Agents are situated in a dynamic environment through which they interact. They are characterized by limited (possibly no) representation of themselves, of the others and of the environment. Their behaviors are based upon stimulus-response rules, decision-making is based on limited information about the environment and on limited internal states and it does not refer to explicit deliberation. In such systems, the individuals do not have an explicit representation of the collective task to be achieved because of their simplicity. Therefore, the solution of the problem is the consequence of successive interactions between agents through environment.

Such a paradigm is convenient to design artificial self-organized systems because it enables the production of complex patterns from interaction between simple agents. It provides means of description for agent behavior and interaction and it authorizes thanks to experiments to observe the collective phenomena.

In this article, we explain why experiments is a key point to understand and analyze such artificial self-organized systems. We claim that an experimental approach relies upon software tools. This article describes our current reflexions about what such tools could be and details how we put into practice some of these ideas in order to analyze two artificial self-organized systems we have built.

The remaining of the article is as follow. The next part presents the two major kinds of application of reactive multi-agent systems and works related to experimentation in these systems. A particular model of self organized system and its current implementations are then presented. From these systems, we illustrate the needs for experimentations and provide key ideas for a platform with some technical aspects of its implementation.

II. Problematics and related works

A. Reactive multi-agent systems

Reactive multi-agent systems can be of two types according to the purpose of the system: simulating a collective phenomena or solving a problem by the collectivity.

The multi-agent paradigm can be applied in the simulation of societies composed of a set of simple interacting individuals. Such simulations enlighten the relationships between the individual behaviors and the collective observed phenomena. This is a reason why recently multi-agent simulations have been undertaken in biology to capture the global effect as a consequence of the behavior and interaction of simple individuals. For example, the aim of the MANTA project [5] was to test hypothesis about the way social structures can emerge, such as a division of labor in the case of ants. Multi-agent models have also been used to investigate the activities of colonies of termites or of Honey Bee [6] [7], or to analyze hypotheses about rats specialization [8]. Various applications of such model on several collective phenomena can be found in [9].

The inherent properties of reactive systems make them convenient to collectively solve problems. These systems can easily be adapted to new constraints at run-time. Applications of these artificial self-organized systems are becoming more and more attractive from scene animation in movies [10]; to optimization problems [11], [12]; or flood forecast [13]. This is because their characteristics make them able to adapt dynamically their function or structure to changing conditions without external intervention. A lot of examples of transposition of natural self-organized systems for problem solving can be found in [14]. In that cases, the development of the system needs also experimentss to find the relevant value of parameters that enable efficient solving of the problem.

B. The need for experimentations

Several authors argue the use of an experimental methodology to investigate, validate and understand self organized systems. For example, [15] explicitly refer to the experimental process : "‘*the development of any agent system, however trivial, is essentially a process of experimentation*"’.

Dorigo et al [16], who proposed the famous Ant algorithm that inspired a lot of efficient algorithms in optimization problems, explicitly mention in their conclusion that *"‘what is also missing, ..., is a theory that allows to predict the system behavior as a function of its parameters and of the characteristics of the application domain"’*. Therefore, if prediction of behavior is difficult (indeed impossible) a priori, experimentations of the system and observations a posteriori are useful.

[17] or [18] argue a novel vision of computing and networking system based on several simple interacting units without centralized control. Such a vision calls for novel methods for the design, the development and the validation of them. [17] explicitly mention in their proposal research agenda the need of experiments to explore the behavior of such systems.

[19] and [20] make a step further in his article. They show that even a simple multi-agent system is beyond the scope of formal methods and concludes by the need of experimental methods.

[21] describe the conditions that lead to the requirement of experimental methods to evaluate the behavior of complex agent.

C. Platforms

Several platforms are dedicated to fast design of collective system and to explore collective behavior made up from simple interacting units.

Swarm[1], Starlogo[2], Netlogo[3], or the synchronous engine of madkit[4] are examples of what platforms look like. They provide means to program the agent behavior, to view graphically how agent evolves in the environments. They provide also some tools to visualize evolution of some key information of the system through graphs as histograms, which display the frequency distribution of a variable, and line plots, which show the changes over the time of some pertinent information (for example the evolution according to the execution cycle of the population of agent). These tools propose a more abstract and concise vision about what happens during runtime.

Netlogo platform uses the concept of "‘behavior space"’. The behavior space consists in an editor that enables the definition of scenario for experiments together with the report of some specific statistics about problem parameters. Scenario specification corresponds to the definition of the initial setup (what commands to executes first, what initial values) for the experiments, the different values of parameters with which experiments have to be undertaken (for each considered parameter, the enumeration of its successive values). The editor also enables the definition of the conditions under which to stop experiments and of the measures to do during experiments (for example, the mean of a variable).

However for all of these tools, graphs are the main elements proposed (as far as we know), they are simple of use but are somehow limited, nothing is proposed neither to store data nor to compute advanced statistics.

D. Some remarks

There are a recent but strong stream of researchers who estimate that the study of complex systems such as the artificial self-organized system needs experiments and therefore adequate tools for that purpose together with a methodology.

The remaining of the article describes our current point of view about what components are needed to correctly undertake experiments with artificial self-organized system and detail how we instantiate some of them in the case of the Hamelin project.

III. Hamelin, Hamelinet : some specific cases of artificial self-organized system

In this section we present three self-organized systems and some comments about the needs for their analysis. The first one is a self-organized phenomenon in biology and the two others are artificial one that are Hamelin : the simulation of the corresponding biological phenomenon and Hamelinet : the transposed model for task allocation in a network.

[1]www.swarm.org
[2]http://education.mit.edu/starlogo/
[3]http://ccl.northwestern.edu/netlogo/
[4]www.madkit.org

A. The biological self-organized system

The self-organized phenomenon in biology to study is social differentiation in rat's group in the diving-for-food situation. This situation is a complex social task in which, for a group of 6 rats, the food accessibility is made difficult by progressive immersion of the only path of access to the feeder. This experimental schedule leads to the emergence of a specialization in the group of rats, in two profiles: supplier and non-carrier rat. The non-carrier (a) animals never dive, but get food only by stealing it from the suppliers after fight. The supplier (b) rats dive, bring the food back to the cage and cannot defend the food they carried. So, confronting groups of rats to a situation in which they have an increasing difficulty to reach food leads to the emergence of a social structure. Biologists are interested by assessing some assumptions about individual behaviors and the interactions that can lead to the social differentiation. For instance, is the apparition of some social pattern possible from a set of interacting individuals without any social cognition ?

B. Hamelin : Simulation of the biological self-organized system

In this section we will detail the agent based model Hamelin which has been built to verify the hypotheses above.

1) The model: In this agent-based model a rat is represented by an agent, the feeder and the immersed path by the environment. All rats are reactive agents characterized by internal state and behavioral items. They behave according to stimulus-response rules which make them react to partial perception of their surrounding environment.

The internal state of an agent is characterized by 4 variables which have been proved to have importance during biological experiments. These variables are:

- The strength of the agent s, which stands for its ability to win when it is involved in a fight
- Its anxiety towards water θ corresponding to its reluctance to dive into water.
- Its hunger h which embodies the need for food and constitutes the motivation for the agent.
- The possessed amount of food $Food$ implemented as the size of the owned pellet.

The activity of the agent is a combination of three actions, to dive, to attack (and fight) and to eat. Each of the actions has a length respectively ld, lf, le and is stochastically triggered or carried out. The associated probability is computed according to the internal state of the rat and biological observations. The dive action is modeled as a response threshold [22]. The fight is modeled as a dominance relationship [23]. We simply reused existing models and coupled them. When the action is effectively performed a reinforcement alters the internal state of the agents allowing them to learn and modify their behaviors according to their past actions [8].

The environment represents the environmental setup. It is characterized by the length of the immersed corridor and by the size of the pellets in the feeder. The characteristics are implemented through two variables: τ the time needed to eat entirely a pellet and η the energy absorbed during a time step of consuming of the pellet. The energy contained in a new pellet is thus $\tau\eta$.

2) The simulator : The kernel was first built as a cyclic simulator which allowed to reproduce with a relative simplicity the biological phenomenon in a synchronous way. The second simulator is an event based one and take into account time dependent mechanisms in a better way. The inputs of these simulators represented the experimental conditions, that are the initial parameters for agents description (s, θ, h, $Food$, ld, ...), the parameters for environment description(η, τ), duration of the experiment (number of cycles or experiment length) and number of agents. It's about a dozen of parameters. The outputs of the simulator are series of values that represent evolution of variables like anxiety of rats during experiment.

C. Hamelinet : a self-organized system for task allocation

The general framework to transpose the Hamelin model consists in a dynamic task allocation problem among machines, connected together in a network. Initially the tasks are available on a central server. The machines can acquire the data by accessing directly the server or by 'attacking' each other. As some policies are put on the server in order to avoid crash, some agents can easily access the server while other not.

This toy example is dedicated to assess the transposition principles of the Hamelin model. The aim of the self organized process is to reduce the exploitation of the network between the machines and the server by means of specialization among machines. It corresponds to dynamically (and efficiently) allocate tasks on an unknown set of machines by making some of them accessing directly the server (because it is easier for them) while others acquire data indirectly (as shown in figure 1.

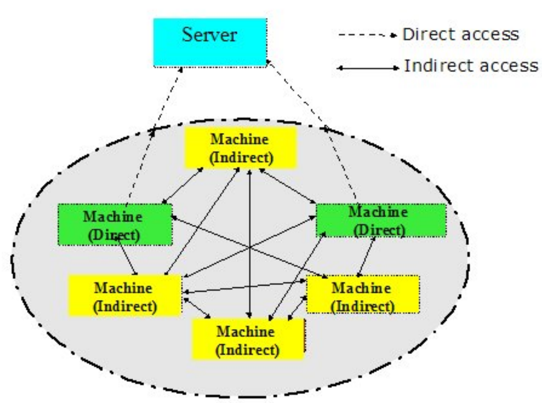

Fig. 1. Expected organization for task allocation problem in network

As we first want to assess the transposition principles, we only need to develop a first prototype at applicative layer of the network that assumes an existing communication architectures and that only deals with the data processing (including execution and redirection). This is the reason why a new simulator has been developed from the Hamelin one by including the server, the machine and the network and why we do not use a network simulator, nor a real network infrastructure.

D. Comments

The aim of Hamelin was first to reproduce the biological phenomenon of differentiation, and secondly to better understand the behavior of the system. The simulation presents a lot of interesting dynamic properties, among which the ability to adapt itself to the number of agents or to evolve according to environmental constraints (for example, the ratio between carrier agents and supplier agents number evolves according to the energetic supply coefficient of a pellet).

From task allocation point of view the aim was to verify that Hamelinet kept the dynamical properties of the Hamelin model. We got some encouraging results like specialization appearance and some improvements in processing time. But these results were obtained with specific instances of the problem and with hand tuned parameters.

Theses results can only be obtained by trial-error experiments in an iterative process since exact behavior of these systems could only be known 'a posteriori'. In order to understand influences of either the parameters or the combinations of parameters, differential analysis are required and a lot of experiments are carried out. One experiment must be proceed a lot of times in order to be statistically valid. So it is useful to store a full and detailed review of preceding experiments and often to analyze data from previous experiment with multiple other views.

To go deeper in the understanding of both the biological model and its transposition we start the development of a platform in order to analyze them through experimental investigations. We develop a software package to test various scenarios in a batch or interactive mode, to collect possibly huge data from experimentation and to interpret them.

IV. Key ideas for analysis artificial self-organized system

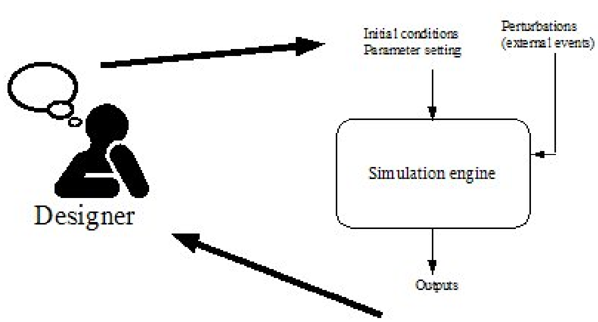

Fig. 2. The use of a simulator

The main aim of a platform is to provide tools for system designers that help them in the analysis of their systems. Analyzing a system includes (from simpler to more complex) the following activities :

- studying under which conditions the expected behavior occurs or don't. It is the basic questions "'what happens if this model is run with such parameters setting?"'. These studies require to interpret the result of the system execution according to different points of view. This can be done with a simulator (see figure 2) where the designer specify the inputs and interpret them. If new inputs are needed, the designer modifies them and run a new experiment.
- validating the system, that is being able to ensure that the expected result will be produce according to initial conditions. This implies being able to specify the initial conditions of the system as a set of intervals for the parameters values. Such studies help to understand the reproducibility of the phenomenon and in case of problem solving to predict the production of a solution to the problem being asked.

  These two first issues are related to the self organized process and can be studied either for simulation or problem solving. The next one is more related to the quality of the output is thus more dependent of the problem solving domain.
- improving the quality of the output of the system. It implies measuring the efficiency of the system to achieve the task and modifying the parameter setting in order to improve its behavior.

A. Generalities about the platform

The platform we propose is our first attempt to answer the requirements that poses an experimental approach to analyze artificial self-organized systems.

It is composed of a set of components that help the system designer (see figure 3). They are conceived as a set of modules that can be coupled. The system designer can use each of them separately or chaining them to undertake a series of experiments (in a kind of batch mode). Therefore, this implies that the data exchanged between each component to be stored in a compatible format in order to be re-used later.

B. Detail of components

1) Kernel: The core module of the platform is obviously the simulation engine that implements the self organized model. Its input are the agents parameters, the environment and the dynamics of both described by parameters settings. Its outputs are made of any update of relevant information about the system.

For example in the Hamelin model, each agent is defined by 4 internal states and any change of them at runtime can be an output.

Furthermore, the systems we wish to analyze have properties related to the dynamics. To enable the study of them we need to introduce during the system execution some perturbations that will correspond to external events. An external event

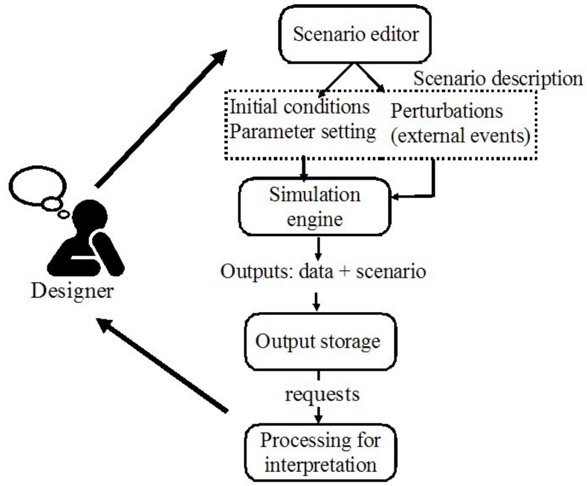

Fig. 3. The platform components

is "‘something"’ that makes the system evolve and which is not modeled in the agents behaviors or environment dynamics (which constitute the internal events of the system).

As agent decisions are stochastic, we have to run several experiments with the same input. This can lead to a mass of data. Moreover, the experiments can be undertaken with different inputs. These considerations lead respectively to the concept of storage of output and scenario editor.

2) Scenario editor: As the kernel as to be run with several different inputs, we define a scenario editor that specifies the conditions of experiments. They are the environmental parameters, the initial value of internal states, the parameters of behaviors, and the number of experiments. For some of them, instead of defining a single value, it is possible to define a set of values. In that case, experiments are executed for each value in the set.

The scenario editor enables also the specification of the perturbations that will occur during experiments. A perturbation is an external event from the simulation engine point of view, that is something that occurs at runtime, which is not due to the "‘normal"’ evolution process of the system and that modifies some of the parameters of the system. It can correspond :

- to a change in the value of a parameter of the environment (for example in Hamelin, a change of the energetic supply coefficient),
- to a change in the value of a parameter of an agent or of all the agents (for example in Hamelin, an increase of the hunger or an increase of the speed of increase of the hunger)
- to a change in the elements of the scenario adding or removing one or more agents or a new type of agent, or a new configuration of the environment, etc. For example, in Hamelinet, it can correspond to a change in network topology (a link between two machines is broken).

Additionally, this editor enables the specification of information related to the storage of produced data and of scenario description files (paths, ...).

3) Storage of outputs: The aim is to manage the mass of information, which constitutes the outputs of the kernel. The principle we follow is to separate the data provided by the execution of the model and their processing for interpretation. Experiments produce raw results (eventually they can be filtered to limit the amount of data), that is they are not processed: no mean, min or max are computed. Even if such a choice can conduct to a vast amount of data, it does not limit the kind of interpretation that can be conducted a posteriori. Furthermore, in order to enable coherent interpretation and further gathering of data (for example all the data obtained with p rats) we store also information that enables to retrieve the scenario description.

A database management system connected to the kernel can achieve these requirements. It authorizes to store information in a structured manner, and information retrieval according to structured requests. Alternatively, some information can be stored as text files in order to have a simple archiving.

4) Interpretation of data / points of view: The mass of information provided by experiments has to be interpreted. Interpretation can be achieved with various ways from plots of a single runs to a study of the mean behavior of a set of different experiments. Statistical tools can help in that interpretation: they provide several tests to analyze what happens in a population of individuals, to measure some relationships between two variables, etc. The inputs of these tools can be raw data from the database or can be the results of a complex request that compute more abstract data from a set of experiments.

C. Technical details of components

These concepts has been partially implemented through two multi-agent models; the first is the Hamelin model for the simulation of collective behavior among rats groups, the second (Hamelinet) deals with the transposition of the first in a toy problem in networks. The overall integration is made in Java.

Currently, in Hamelin case, the scenario editor allows the specification of the initial parameters of the agents behaviors, of the environment constant (amount of hunger increase per cycle, energetic supply, etc). The values can be provided as iteration pattern : *6 10 2* iterates from value 6 to value 10 by a step of 2. In Hamelinet case, we developed a "‘network builder"’, that is a tool that enables to specify a network topology in order to check the model behavior in specific configurations. Furthermore, each scenario specification can be stored (as text files) if requested and reloaded.

Currently, we can not specify perturbations: neither a rapid change in a parameter value or in environment (conection down), nor the introduction of a new agent (or its death).

As mention above, the simulation engine is based on a discrete event simulator programmed in Java. Currently its invocation is done by specifying the parameters values and the number of experiments. In Hamelin, iterations on parameter values are managed from the scenario editor interface.

Outputs of the simulation engines can be stored as text files. These files are organized in an hierarchy in order to easily retrieve data from a given experimentation (a set of parameter values). However, if text files can be useful to fast (and simple) analyzes, they are obviously inadequate when analyzes request comparison of data from several experiments. It is why we use a database management system (MySql+ EasyPhp) that allows storage of data and complex retrieval requests. Furthermore, the EasyPhp interface authorizes direct requests to the database.

The analyzes of data can be made for the simplest cases (line plots) with usual spreadsheets tools. To handle more complex analyzes, such as correlation factors, etc; we use statistical tools such as R and SPSS.

V. Examples of use

We present here how we are currently using our platform in order to analyze our systems (Simulation or problem solving).

The first (and basic) case of use is to verify the appearance of profiles in application case or in simulation one.. A first manner to undertake it is to run an experiment and plot the evolution of hunger, anxiety and strength of the rats being involved in the experiments. Concretely, to achieve that

1) we specify a basic scenario (make ten runs with the same parameters),
2) we do experiments and store evolution of pertinent parameter in text files,
3) we do plotswith a spreadsheet or a statistical tool.

This simple assessment can be sufficient when parameters are well tuned (specialization very frequently appears). However this use of the system is not enough when we have to assess the frequency of specialization or its quality because it only focuses on the individual level. Thus we need a more collective analysis.

Originally, in biological experiments, 13 variables were computed for each rat in order to describe its behavior. These variables were treated by a factor analysis (Principal Components Analysis) followed by a cluster analysis on the individual coordinates in the first factorial plan.

This is a more convenient point of view to achieve our purpose. Thus, to achieve this collective analysis,

1) we describe various scenario,
2) we run each of them for a fixed number of times and store the raw data in database,
3) we compute these 13 variables from database,
4) we provide them as inputs to a statistical tool that realizes the desired computation (PCA + cluster analysis).
5) according to the interpretation of the results we made, we can have an iterative approach : we can specify new scenario, etc.

In Hamelin, these steps can be undertaken trough the user interface of the system of separately from the operating system interface.

VI. Concluding remarks

There are a recent but strong stream of researchers who estimate that the study of complex systems such as the artificial self-organized ones needs experiments and therefore adequate tools for that purpose together with a methodology.

This article describes our current point of view about what kinds of components are needed to correctly undertake experiments with artificial self-organized systems. It details how we implement or integrate these components in two different directions: simulation and application. As work is under progress, in the application case, the integration of modules is not completely done (some interconnection between them have to be done manually).

We are currently experimenting the platform to analyze the Hamelin model and to assess the relevance of the principles that guide its building. It is a work under progress and we have not yet fully exploited all the different possibilities of the platform, nor of our proposition. A deeper assessment has to be done, however the first uses of the platform confirmed the usefulness of it.

We consider this work as a first step toward an experimental approach for the analysis of artificial self organized systems.

Future works will be undertaken in several directions.

We have to fully implement our proposition by including the possibility in the scenario editor to define perturbations that occur at runtime. The first uses underlined the need to specify filter to apply to the output of the simulation engine in order to prevent the production of huge mass of data.

More generaly, our reflection concerns the introduction into our platform of an optimization component that closes the loop (see figure 4 and allows a complete automation of the improvement of the model. Such a principle was tested in [24]) and it seems to be of great interest in the design of artificial self-organized systems.

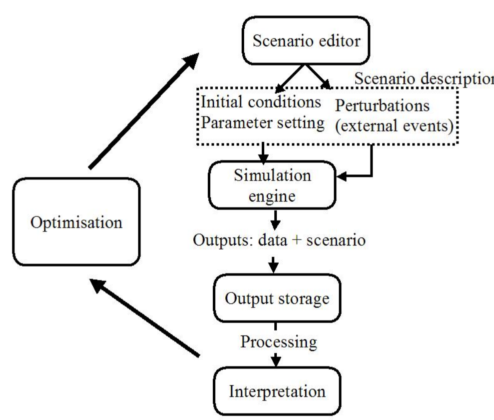

Fig. 4. A proposition for automatic tuning of the parameters

Acknowledgment

This work was partially funded by the Region Lorraine in the Pole de Recherche Scientifique et Technologique ”Intelligence Logicielle”. We thank Sophie Schumacher for their contributions to the presented work.

References

[1] Camazine S., Deneubourg J.-L., Franks N., Sneyd J., Theraulaz G., Bonabeau E. *Self-Organization in Biological Systems.* Princeton University Press, 2001.

[2] S. Kazadi, M. Chung, B. Lee and R. Cho *On the Dynamics of Puck Clustering Systems*, Robotics and Autonomous Systems, 46(1), pp. 1-27, 2004.

[3] G. Weiss, *Multiagent systems a modern approach to distributed artificial intelligence*, Mit Press 1999

[4] Van Parunak H. *Go to the Ant: Engineering Principles from Natural Multi-Agent Systems*, Annals of Operations Research 75, pp 69-101, 1997.

[5] B. Corbara, A. Drogoul, D. Fresneau and S. Lalande, *Simulating the Sociogenesis Process in Ant Colonies with MANTA* in Towards a Practice of Autonomous Systems II, MIT Press, Cambridge,1993.

[6] O. Miramontes , DeSouza O. *The Nonlinear Dynamics of Survival and Social Facilitation in Termites*. Journal of Theoretical Biology, Vol. 181, No. 4, Aug, pp. 373-380, 1996

[7] D. Sumpter, D. Broomhead, *Formalising the link between worker and society in honey bee colonies*, First International Workshop, MABS 98 LNAI N1534, pp95-110.

[8] V. Thomas C. Bourjot V. Chevrier, D. Desor. *Hamelin: A model for collective adaptation based on internal stimuli.* From animal to animats 8 - Eighth International Conference on the Simulation of Adaptive Behaviour 2004 - SAB'04, Los Angeles, USA, pp 425-434, 2004.

[9] *Individual-Based Models an annotated list of links* http://www.red3d.com/cwr/ibm.html

[10] *MASSIVE Multiple Agent Simulation System In Virtual Environnement* - http://www.massivesoftware.com

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

[12] A. Dury, F. Le Ber and V. Chevrier, *A reactive approach for solving constraint satisfaction problems: assigning land use to farming territories.*, Lecture notes in Artificial intelligence, Intelligent Agents V , n 1555, J.P. Muller, M.P. Singh et A. S. Rao (eds), pp397-412, Springer, 99.

[13] J. P. George , M. P. Gleizes, P. Glize and C. Regis, *Real-time Simulation for Flood Forecast: an Adaptive Multi-Agent System STAFF* - Proceedings of the AISB'03 symposium on Adaptive Agents and Multi-Agent Systems, University of Wales, Aberystwyth, 7-11 April 2003.

[14] G. Di Marzo Serugendo, A. Karageorgos, O.F. Rana and F. Zambonelli (Eds) *Engineering Self-Organising Systems: Nature-Inspired Approaches to Software Engineering* LNAI 2977, 2004.

[15] M. Wooldridge and N. R. Jennings, *Pitfalls of agent-oriented development*. In Proceedings of the 2nd International Conference on Autonomous Agents, pages 385391, 1998.

[16] M. Dorigo, E. Bonabeau and G. Theraulaz, *Ant algorithms and stigmergy*, in Future Generation Computer Systems, Volume 16, Number 8, pp851–871, June, 2000.

[17] F. Zambonelli, M.P. Gleizes, M. Mamei and R. Tolksdorf, *Spray Computers: Frontiers of Self-Organization*, extented version of 1st International Conference on Autonomic Computing, New York (NY), May 2004.

[18] F. Zambonelli and V. Parunak, *Towards a Paradigm Change in Computer Science and Software Engineering: a Synthesis*, The Knowledge Engineering Review, 2004, to appear.

[19] B. Edmonds *Using the Experimental Method to Produce Reliable Self-Organised Systems*, The 2nd International Workshop on Engineering Self-Organising Applications (ESOA 2004) at 3rd AAMAS, New York, July 2004.

[20] B. Edmonds and J. Bryson *The Insufficiency of Formal Design Methods -The Necessity of an Experimental Approach for the Understanding and Control of Complex MAS*, (936-943) AAMAS 04.

[21] J. Bryson, W. Lowey and L. A. Stein, *Hypothesis Testing for Complex Agents*, in the proceedings of the NIST Workshop on Performance Metrics for Intelligent Systems, August, 2000.

[22] G. Theraulaz, E. Bonabeau, J.L. Deneubourg, *Response threshold reinforcement and division of labour in insect societies*, Proc. Roy. Soc. London B 265, pp327-332, 1998.

[23] C. K. Hemelrijk, *Dominance interactions, spatial dynamics and emergent reciprocity in a virtual world*. Proceedings of the fourth international conference on simulation of adaptive behavior, vol. 4, pp. 545-552. Cambridge, MA: The MIT Press, 1996.

[24] B. Bertrand, *Adaptation des parametres dans les systemes multi-agents reactifs :Application en robotique mobile*, Rapport de DEA, Univ Henri Poincare, Nancy 1, 2004 (in french).