

Comparative Study of Self-organizing Robotic Systems Regarding Basic Architecture

Irina-Gabriela Lolu, Aurelian Mihai Stanescu

► **To cite this version:**

Irina-Gabriela Lolu, Aurelian Mihai Stanescu. Comparative Study of Self-organizing Robotic Systems Regarding Basic Architecture. Luis M. Camarinha-Matos; Pedro Pereira; Luis Ribeiro. First IFIP WG 5.5/SOCOLNET Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS), Feb 2010, Costa de Caparica, Portugal. Springer, IFIP Advances in Information and Communication Technology, AICT-314, pp.177-184, 2010, Emerging Trends in Technological Innovation. <10.1007/978-3-642-11628-5_19>. <hal-01060801>

HAL Id: hal-01060801

<https://hal.inria.fr/hal-01060801>

Submitted on 16 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Comparative Study of Self-Organizing Robotic Systems Regarding Basic Architecture

Irina-Gabriela Lolu and Aurelian Mihai Stanescu

Faculty of Automatic Control and Computers,
University POLITEHNICA of Bucharest, Romania
lolu_irina@yahoo.com, ams@cpru.pub.ro

Abstract. Self-organization has become an intensely researched area, being considered the new control system science. Different self-organizing algorithms (bio and non-bio inspired) have been developed by scientist worldwide with general applicability in autonomous systems, multi-robot systems, autonomic networking, sensor and actor networks and mobile ad hoc networks. This paper presents a comparative analysis of basic architectures for multi-robot systems from the point of view of self-organization capabilities. We identify requirements for implementing efficiently self-organization mechanisms with high applicability in multi-robot systems.

Keywords: self-organization, robotic systems, architecture

1 Introduction

Nature has always inspired scientists to develop techniques, algorithms and mechanisms and to apply them in artificial systems. This is also the case of self-organization; inspired from biology, physics and chemistry and used in computer science and robotics. This paper presents a critical analysis of basic architectures for robotic systems from the point of view of self-organization capabilities. We also propose a simple architecture that enables efficient implementation of self-organizing mechanisms.

Since it was introduced in 1947 by W. Ross Ashby[1], self-organization has caught the interest of the research community in various fields (ex. autonomous systems, autonomic networking, sensor and actor networks, mobile ad hoc networks), including robotics. The generally accepted definition for self-organization is: the mechanism enabling a system to change its internal organization according to environmental changes without explicit external command.

There are two approaches in self-organization: stigmergy and social. The concept of stigmergy was introduced by Grassé [2] to explain the social behavior of termites. It was demonstrated that coordination in ants and termites colonies is achieved through indirect communication by the means of a chemical substance, pheromone, deposited in the environment. The important characteristics of pheromones that have to be kept in mind when implementing stigmergy are: they evaporate in time (evaporation), the concentration increases when new pheromones are released at a marked location (aggregation) and they propagate through environment (propagation).

The bio-inspired stigmergy mechanisms aim to achieve self-organization through simple rules and interactions between robots/agents, using indirect communication by the means of artificial "pheromones". Stigmergy has been used in applications for hierarchical task networks[3], manufacturing control [4], coordination of unmanned vehicles[5] a.s.o.

In contrast with stigmergy, the social approach uses direct communication. These techniques are inspired from natural social behaviors encountered in human societies, business organizations and even economic systems. The most popular social self-organization mechanisms are market-based (in which entities act to increase their profit)[8] and delegation based mechanisms (in which delegation is based on trust or reputation level, authority and voting) [9].

2 Contributions to Technological Innovation

The adaptive and robust self-organizing systems are an increasing trend nowadays, especially in software application, although the developers have to face great challenges: non-linearity, unpredictability, instability and sensitivity. When designing a self-organizing robotic system the challenges are even more difficult due to hardware limitations and time restrictions. Our work is a contribution to the design of such systems and concerns with one of the first steps of developing a robotic system: choosing the basic architecture. We are contributing with an analysis of existing architectures and we also propose a new architecture that ensures the implementation of different types of self-organizing mechanisms. We hope this paper will encourage further research in self-organizing robotic systems, enabling the development of the future robot societies.

3 Comparative Analysis of Basic Architectures

In this chapter we will discuss several robot architectures from the point of view of self-organization capabilities. Even though there are essential differences between stigmergy and the social approach, both have the same following properties:

- Absence of external control
- Decentralization
- Emergence
- Self-maintenance
- Self-building
- Adaptive
- Sensitivity
- Low predictability
- Robustness.

There are two different types of implementation for stigmergy: software and hardware. The software implementation has no special requirements (small processing power, shared memory for environment simulation), but the hardware

implementation is restricted by the physical proprieties of pheromones. Still some solutions have been proposed by Johansson and Saffiotti [6], [7].

The social techniques are agent-oriented, thus the basic requirement is an agent-oriented architecture. Therefore we will concentrate on studying architectures based on multi-agent systems (MAS). For efficient implementation we also consider the following requirements: fast, long distance communication capabilities and considerably more processing power and memory than necessary for implementing stigmergy.

While the social approach is mostly used in more complex heterogeneous systems that require hierarchy, social learning, and knowledge propagation through gossiping [10], the stigmergy approach has some important advantages:

- Simplicity (robots/agents don't need a complex deliberative mechanism to implement the simple rules of stigmergy)
- Asynchrony (robots/agents act asynchronous according to the information found in the environment)
- Anonymity(robots/agents are not aware of each other)
- Public knowledge (all knowledge is public and can be found in the environment)
- Low –cost

We are considering the following criteria for analyzing existing MAS architectures for robotic systems: coordination – centralized/decentralized, resource requirements and communication capabilities.

There have been designed architectures for self-organizing multi-robot systems, but they implement a simple, specific self-organization mechanism using none or very little direct communication. From this category we mention ALLIANCE [11], Cebot[12], SWARM-bots[13] and ABBA[14]. These architectures cannot be used for developing more complex applications that require a social self-organization mechanism demanding high-level, message-oriented communication.

The first category of architectures that we present is the layered architectures category, depicted in Figure 1. These architectures have two [15], three [16] or even eight layers [17], that can communicate only with the neighboring layers using an inter-layer communication protocol (ex. TCP/IP). The drawback of this architecture is that the superior layers have to use the intermediate layers to communicate with the inferior layers, causing latencies which could be significant in real-time applications.

The top layer is usually called the Cognitive Layer and is represented by one or more deliberative agents that could enable self-organization, if chosen properly. These architectures were designed for cooperative robots, but were not meant especially for self-organization mechanisms, thus even if they have a multi-agent system these agents were not chosen in a way that enables emergent behavior.

Most of the architectures have a single agent that coordinates and controls the other agents (“Global Manager” or “Coordinator” [15],[18]), which means centralizing the decision, contrary to the distributed coordination essential for self-organization. Table 1 summarizes the studied architectures.

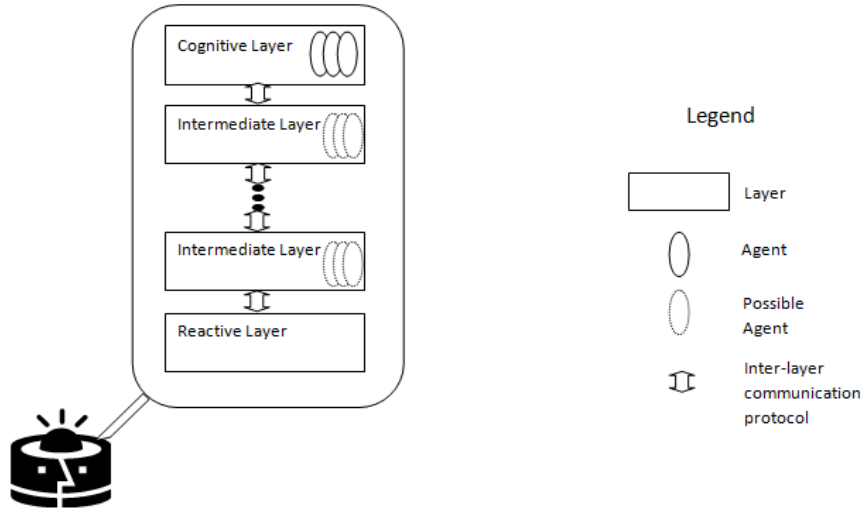


Fig. 1. Generic layered architecture.

Another category of architectures are the architectures with one layer composed of different types of agents: Body Agents, Deliberative Agents, Service Agents, Behavioral Agents a.s.o.(Figure 2) We mention here the ARMADiCO[20] architecture which has distributed coordination among agents (based on a utility function) that could enable emergent behavior.

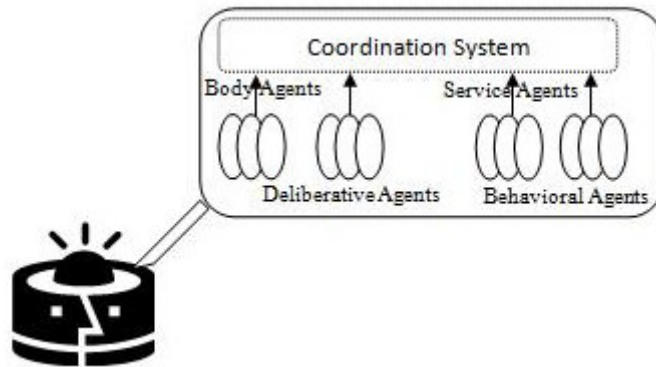


Fig. 2. One layer architecture based on multi-agent system.

Table 1. Layered architectures examples.

Name	No. of layers	Description	Disadvantages
Physical Robot Agent [15]	2	<ul style="list-style-type: none"> • Action Layer: Executor, Repository of Tasks and State Monitor • Cognitive Layer: Decision Maker, Coordinator and Negotiator 	<ul style="list-style-type: none"> • Centralized control
Acromovi [19]	4	<ul style="list-style-type: none"> • Applications and Middleware Layer - multi-agent systems implemented in Java • ARIA and Saphira Layers – implemented in C++ 	<ul style="list-style-type: none"> • Inter-layer communication – Java JNI and Java RMI • High resource requirements
ARA [16]	3	<ul style="list-style-type: none"> • Reflexive Layer – on the robot • Reactive and Cognitive Layers - on another machine • High-level behaviors composed of elementary behaviors 	<ul style="list-style-type: none"> • Separation on two machines
Busquets [18]	2	<ul style="list-style-type: none"> • Executive and Deliberative systems • Coordination through bidding executed by a Coordinator 	<ul style="list-style-type: none"> • Centralized control

4 An Architecture for Self-Organizing Robotic Systems

The disadvantages we identified in our analysis of the above architectures motivated us to design a simple, flexible architecture, with low resource requirements and decentralized control. Therefore we propose the two layered architecture with simple inter-layer communication (plain function call) depicted in Figure 3. Removing the intermediate layers we eliminate message passing, we reduce inter-layer communication (by embedding the intermediate layers in the superior or inferior layers), although at the cost of increasing intra-layer communication.

The Action Layer is a simple implementation of tasks (reactive) and its design does not influence the self-organization capabilities, but it has an impact over the performance of the system. The top layer (Cognitive Layer) is a multi-agent system with three or more agents, depending on the self-organization mechanism we want to implement. This ensures the flexibility of the architecture.

The DF and AMS agents implement the analogue components defined in the FIPA standard [21], and they are necessary to ensure self-organization in a multi-robot system. The DF agent is responsible for maintaining a list of all agents and their

services in the system and offers the following services: updating and search. The AMS agent is responsible for registering and unregistering agents.

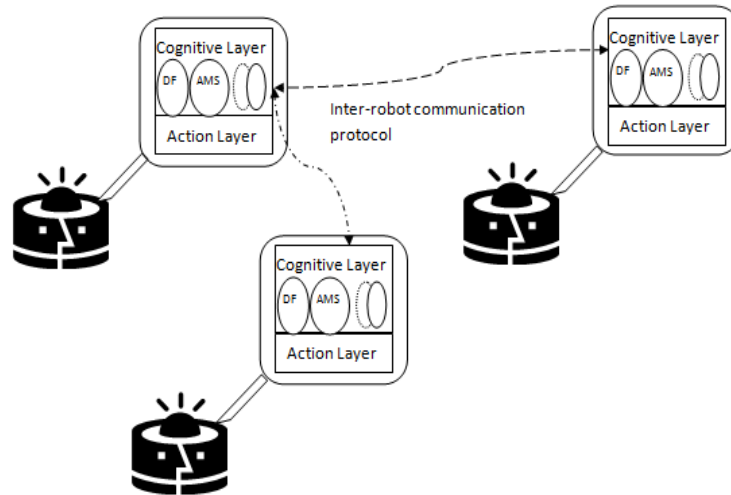


Fig. 3. A two layered architecture for self-organizing multi-robot systems.

Choosing the other agents depends directly on the application and on the robots capabilities. In our opinion there is no need for “body” agents corresponding to sensors, but agents that control actuators could be useful for resolving resource concurrency. An example for emergent behavior that could be implemented using this architecture is navigation with obstacle avoidance. In this case we must define the following agents: motor control agent – controls the motors and decides which agent has priority, move forward agent – moves forward the robot, avoid obstacle agent – when sensing an obstacle stops the robot and changes the direction of the robot.

Our current implementation of this architecture is on a Lego NXT robot (we wanted to underline the advantages of this architecture by implementing it on a machine with low computational power), using leJOS NXJ (a Java programming environment). We implemented a simple agent platform that uses shared memory for local communication and Bluetooth communication for inter-robot communication (a Bluetooth Agent). In our system every robot has an agent platform and the agents have the possibility of registering on multiple platforms (when agents communicate they use the local platform’s communication service). At the moment we are developing a practical application that gives the possibility to test different self-organization algorithms in different scenarios in order to have consistent results.

5 Conclusions

In this paper we presented our analysis of the studied architectures for self-organizing robotic systems with respect to the identified requirements for the implementation of

self-organizing mechanisms. The problems we encountered are: centralized coordination, high resource requirements and inter-layer slow communication protocols. Therefore we proposed an architecture that has none of these disadvantages. We intend to further develop this architecture and test it with different self-organization mechanisms in our further work.

References

1. Ashby, W.R.: Principles of the Self-Organizing Dynamic System, In: Journal of General Psychology 1947. volume 37, pp 125--128 (1947)
2. Grassé, P.P.: La reconstruction du nid et les coordinations interindividuelles chez *Bellicositermes natalensis* et *Cubitermes* sp., La théorie de la stigmergie : essais d'interprétation du comportement des termites constructeurs, *Insectes Sociaux.*, 6, pp. 41--84 (1959)
3. Parunak, H. V., Belding, T., Bisson, R., Brueckner, S., Downs, E., Hilscher, R.: Stigmergic Reasoning over Hierarchical Task Networks In: Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009), Decker, Sichman, Sierra and Castelfranchi (eds.), pp. 1195--1196 (2009)
4. Karuna, H., Valckenaers, P., Saint-Germain, B., Verstraete, P., Zamfirescu, C. B., Van Brussels, H.: 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)
5. Parunak, H. V., Brueckner, S., Sauter, J.A.: 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)
6. Johansson, R., Saffiotti, A.: Navigating by Stigmergy: A Realization on an RFID Floor for Minimalistic Robots. In: Proc of the IEEE Int Conf on Robotics and Automation (ICRA), Koba, Japan (2009)
7. Edelen, M. R.: Swarm intelligence and stigmergy: Robotic implementation of foraging behavior. Master of Science thesis, University of Maryland, College Park, Maryland, USA (2003).
8. Hassas, S. , Castelfranchi, C., Marzo, D., Karageorgos, A.: Self-organizing Mechanisms from Social and Business/Economics Approaches, *Informatica*, Vol. 30, No. 1. (2006)
9. Schillo, M., Fley, B., Florian, M., Hillebrandt, F., Hinck, D.: Self-organization in multiagent systems: from agent interaction to agent organization, In: Proceedings of the 3rd International Workshop on Modeling Artificial Societies and Hybrid Organizations (MASHO'02), Workshop at KI2002, the 25th German Conference on Artificial Intelligence Aachen, pp. 47--56 (2002)
10. Jelasity, M.: Engineering emergence through gossip. In: Edmonds, B., Gilbert, N., Gustafson, S., Hales, D., Krasnogor, N. (eds.), Proceedings of the Joint Symposium on Socially-Inspired Computing, pp 123--126, Hatfield (2005)
11. Parker, L. E.: ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Cooperation, *IEEE Transactions on Robotics and Automation*, 14 (2) (1998)
12. Fukuda, T., Kawachi, G.: Cellular robotic system (CEBOT) as one of the realization of self-organizing intelligent universal manipulator, In: Proceedings of the 1990 IEEE Conference on Robotics and Automation, pp. 662--667 (1990)

13. Dorigo, M., Trianni, V., Şahin, E. , Groß, R., Labella, T. H., Baldassarre, G., Nolfi, S., Deneubourg, J.-L., Mondada, F., Floreano, D., Gambardella, L. M.: Evolving self-organizing behaviors for a swarm-bot. *Autonomous Robots*, 17(2–3) pp. 223--245 (2004)
14. Jung, D., Zelinsky, A.: An architecture for Distributed Cooperative Planning in a Behavior-Based Multi-Robot System, In: *Journal of Robots and Autonomous Systems*, 26(2-3) pp. 149-174 (1999)
15. Eze, J.; Ghenniwa, H.; Shen, W.; “Distributed Control Architecture for Collaborative Physical Robot Agents” IEEE International Conference on Systems, Man & Cybernetics, Washington DC, pp 2977--2982 (2003)
16. Neves, M. C., Oliveira, E.: A multi-agent approach for a mobile robot control system, In: *Proceedings of Workshop on “Multi-Agent Systems: Theory and Applications”*, pp. 1--14 (1997)
17. Shaw, M., Garlan, D.: *Software Architecture: Perspectives on an Emerging Discipline*. Prentice Hall (1996)
18. Busquets, D., Sierra, C., López de Mántaras, R.: A multiagent approach to qualitative landmark-based navigation, *Autonomous Robots*, vol.15, pp. 129--154 (2003)
19. Nebot, P., Cervera, E.: Agent-based Application Framework for Multiple Mobile Robots Cooperation, In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1521--1526, Barcelona, (2005)
20. Innocenti, B., Lopez, B., Salvi, J.: Resource coordination deployment for physical agents, In: *6th Int. Workshop of AAMAS: From Agent Theory to Agent Implementation*, pp.101--108 (2008)
21. The Foundation of Intelligent Physical Agents, <http://www.fipa.org>