Emergence in Collective Robotics: A Case Study

Soraya Mostéfaoui
Department of Informatics, University of Fribourg, Switzerland

Abstract: *In this paper we propose a robotics framework for studying emergence and collective adaptation. We describe two sets of experiments, where a pool of heterogeneous Khepera robots, participate in adapting the collective behavior, without being aware of it but just in virtue of the design of their internal structure. The innovative aspect in our approach rests on a system integrating communication as an active and dynamic component in the adaptation, and not only as a static part of the robots interactions.*

Keywords: *Emergence, autonomous agents, collective robotics, communication, adaptation.*

Received February 19, 2003; accepted May 19, 2003

1. Introduction and Motivation

Our work fits in the framework of *autonomous agents*. We are concerned with collective phenomena (behaviors) and their issues, and more precisely the way to carry out solutions that allow a heterogeneous communicating multi-agent (multi-robot) system to adapt the collective behavior, in face of a changing environment. This problem is common and is still given a lot of concern in multi-agent systems communities. However the way we solve the problem is quiet uncommon. The innovative aspect of our approach rests indeed on a system integrating a dynamic form of communication, which provides a flexible, much richer and more life-like communicating systems.

Our work is supported by two types of experiments, namely those involving multi-agent simulations and those involving real robots. Our motivation of using robots to study multi agent systems (MAS) is twofold. First we provide experimental setup where agents are *situated* and *embodied* [15]. This means that robot models do not deal with abstract descriptions, but with noisy and changing environment. Moreover, robots have bodies and experience the environment directly but imperfectly, and their actions with the environment are part of a dynamic system (see [12] for a complete reference to embodied cognitive science). Second, robots can be viewed as autonomous agents that act independently without any guidance from the user.

The rest of this paper is organized as follows. Section 2 gives a short overview of the state of the art in autonomous agents and related areas, thus allowing the reader to be given the basic key concepts and to understand the framework within which our work comes. Section 3 gives the aim of our work and describes our system. The materials and methods used in our experiments are presented in section 4. Section 5 describes the two sets of experiments that have been implemented. Section 6 concludes this paper with a short summary of the results of our studies.

2. Background and Related Work

This research draws from several fields, including multi-agent systems, behavior-based robot control and emergent systems. A brief review of the most relevant work follows.

2.1. What Are Agents?

Agents can simply be defined[1] as computer systems (entities), that are capable of autonomous actions in some environment in order to meet their design objectives [6]. An agent will typically sense its environment by physical sensors in the case of agents situated in part of the real world like for instance robots, or by software sensors in the case of software agents, and will have available a repertoire of actions that can be executed to modify the environment [17, 19].

2.2. Multi-Agent Systems

A Multi agent system is a system where multiple agents coexist in a common environment. The central idea of MAS is that agents may cooperate (work together), or compete (against each other), to solve problems that are beyond the individual capabilities or knowledge of each agent, but at whole bring fourth a collective behavior [3, 5]. It is this collective behavior that will be the focus of the research. This idea is simply advocated by Marvin Minsky in [10], *"as agents are the members of a population that together produce a behaving system with motives"*. What is attempted here is the application of a reduction principle on the individual behaviors in the sense of the

[1] Note that this is not a formal definition.

question *"what elemental behaviors can be brought together in interaction for some interesting meta-level behavior to be generated?"*

The inspiration of this approach results from the observation of natural systems that relay on some collective behavior to successfully achieve autonomy. Typical often-cited examples of such systems are anthills, beehives or termite colonies.

2.3. Robots as Intelligent Agents

Robotics is particularly a good domain for studying MAS. Robots can be viewed as agents enjoying the following characteristics:

- *Embodiment:* robots have bodies and experience the world directly [2]. Their actions are part of a dynamic with the world and have immediate feedback on the robots.
- *Situatedness:* the robots are situated in the world [15], they do not deal with abstract descriptions, but with the *"here"* and *"now"* of the environment that directly influences the behavior of the system.
- *Autonomy:* robots operate without the direct intervention of humans or others, and have control over their actions and internal states [18].
- *Social ability:* robots can communicate via some kind of robot-communication languages, and typically have the ability to engage in social activities in order to achieve their goals.
- *Mobility:* robots are mobile; this mobility provides a simple abstraction for a complex distributed system.
- *Adaptability:* robots can adapt their individual behaviors, or the collective behaviors to new situations, while in a continuous interaction with the environment.

2.4. Emergent Systems

"Emergence is first of all defined as the creation of new properties" Liyod Morgan

A designer or observer[2] of MAS may call a behavior emergent when this behavior cannot easily be deducted from the individual properties of the agents of the system, or when these properties are not readily accessible. Emergence arises from the interactions of the agents in the system, in the same way as the chemical properties called viscosity or fluidity *"emerge"* from the physical properties of the molecules composing the liquid. The notion of emergence is an epistemological one, a property of a collective emerges from the individual properties when the most adequate tools used to study the individuals are not the same as those used to study the collective [13].

An emergent system has three important characteristics: first, it can accomplish complex tasks with little and simple individual behaviors. Secondly, a change in the environment may influence the same system to generate a different task or structure, without any change in the behavioral characteristics. Finally, any small differences in individual behaviors can influence the collective behavior of the system. Therefore, social complexity of the system is compatible with simple and identical individuals, as long as the communication among the members can provide the necessary amplifying mechanism i.e. allows the needed communications.

3. Aim of the Work

Our application tackles a common problem in collective robotics, which is the study of emergence in dynamic environments, using a set of communicating robots[3]. This problem is still given a lot of concern in MAS communities. However the way we solve the problem is quiet uncommon. The innovative aspect of our approach rests indeed on a system integrating a dynamic form of communication. In one side, we consider communication as an active component in the collective adaptation, not only as a simple static part of the robot interactions with the environment but also as a part of the robot behavior and thereby as an inter-environmental entity. In the other side, communication in our model is dynamic, and is affected by any extra- and inter- environmental changes.

This work relates to other works in the framework of emergent MAS and collective robotics, in which communication remains static during all the lifetime of the system, and is, simply viewed as part of the robot interactions. However the major contribution of our work in relation to other works lies in the use of a flexible, much richer and more life-like communication models.

3.1. System Description

To study complex phenomena like emergence, we instantiate the problem to a simpler one, in which the collectivity has a predefined set of behaviors and in which we are concerned with the *"commutation"* of these behaviors, in response to an external event.

Mainly our system has the two following collective behaviors (states) *"dispersion"* and *"agglomeration"*. The task in the first state is to make the robots go away from each other, and in the second state to make them stay together. These two collective behaviors are not known in advance by the robots, and are not embodied into each robot as predefined goals; they are expected to be the outcome generated by the robots interactions and communications. For each of the collective behaviors, corresponds an individual behavior, respectively *"work"* and *"sleep"*, which consist themselves of a set of behavioral rules at the robot controller's level (Figure 1). The general algorithm of our system can be summarized as follows:

Task = Work;

[2] It is vital to emphasize the role of the observer in this definition.

[3] As we focus now on robotics we will use the term robot instead of the term agent.

```
While (true) do
  {
   Task ();
   UpdateNetTopo ();   /* the network topology is
                          updated dynamically */

   If (external event)   /* an external event is
                            perceived */

   REACT ();          /*  react to the event */
  }

Procedure React ()

 {
   Task = Task-1;       /*commute the task  */

   GetNetTopo ();        /* get an updated version of
                            the network topology */

   Propagate ();         /* propagate the signal
                            according to the network */
}
```

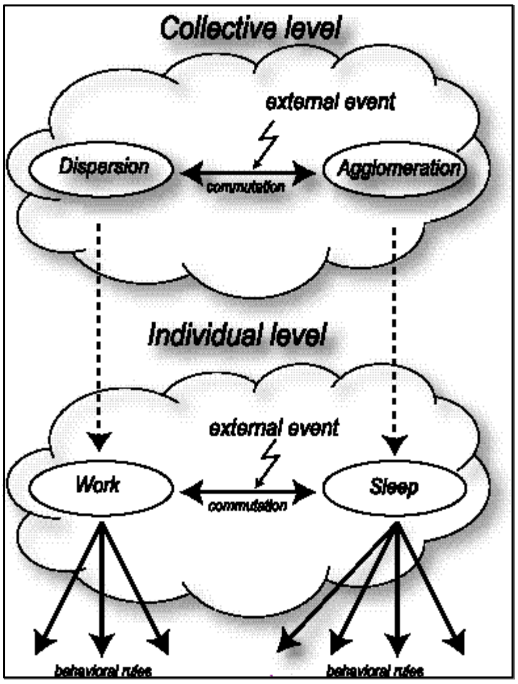

Figure 1. Behaviors commutation.

4. Materials and Methods

4.1. The Robots

For our experiments we use Khepera robots. Khepera is a miniature mobile robot with functionality similar to that of larger robots used in research and education [2, 9, 11]. Khepera was originally designed as a research and teaching tool for a Swiss Research Priority Program at EPFL[4]. Its basic features are as follows: a 32 bit processor running at 16 MHz, an energy autonomy of half an hour, 8 infra-red (IR) sensors (6 on the front of the robot referred to as IR0 to IR5 from left to right, 2 in the back referred to as IR6 and IR7), and two wheels controlled by two DC motors. IR sensors allow Khepera to evaluate the proximity of obstacles on the basis of reflected light of these obstacles (note that this evaluation depends as well on the level of the ambient light). Khepera is a convenient platform for both single- and multi-robot experiments: 20 Kheperas can easily work on a 2 m^2 surface.

4.2. The Communication Devices

A number of researchers have explored heterogeneity at the behavior level, others have instead explored heterogeneity at the hardware level. Our research takes the latter approach in that robots are equipped by different sets of sensors or communication devices, and thereby they have different domain knowledge (Figure 2).

The different devices that were used in our experiments are:

- *Moderok:* Module D'affichage pour le Robot Khepera [16], developed in collaboration with the Ecole d'ingénieurs de Fribourg. This module enables us to display a given pattern by turning-on some light-emitting diodes (LEDs) in a 16×4 array.
- *K213 vision turret* [8], developed by K-team, it contains a linear image sensor and global light intensity detector. The image sensor is an array of 64×1 cells, giving a linear image of 64 pixels with 256 gray-levels each. The optics was designed to bring into focus objects at a distance of 5 to 50 cm in front of the robot.
- *360 K2D video turret* [8], a high-resolution color camera using a spherical mirror for a full 360-degree field of view.
- *Radio turret* [7], a compact radio modem adapted to the Khepera bus, with its own local processor for management of the whole communication procedure, which includes the data encoding, transmission and reception, error detection and correction as well as the support of the protocol with Khepera using the local multiprocessor network. The radio turret makes it possible to communicate with other Kheperas equipped with radio turrets using local path mode, as well as with a radio base station connected to a host computer using global path mode.

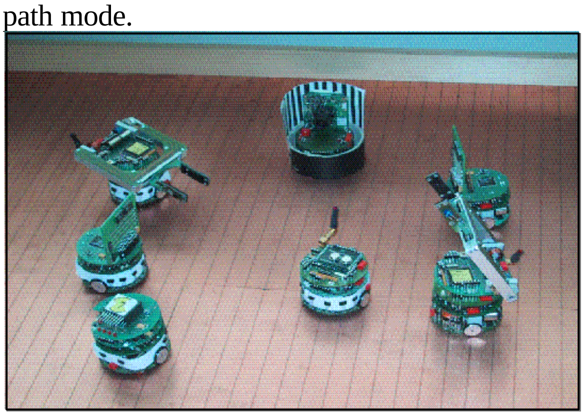

Figure 2. Seven Kheperas equipped with different devices.

[4] Ecole Polythechnique Fédérale de Lausanne.

4.3. The Communication Schema

To fulfill the requirements of our experiments, namely heterogeneous and dynamic communication, and in order to avoid using any positioning or navigating system, we simulate the heterogeneity of the communication devices using the radio turret and a set of appropriate algorithms. Therefore all information is sent by radio, but during the execution different subsets of robots are allowed to communicate, according to the communication devices of the robots and their relative positions.

When robots are moving their relative positions are changing i.e. the distance between them. These movements affect the topology of the communication network. In order to take into account these movements and update the topology of the network in a dynamic way during runtime, we first assign to each robot an ID and a pair of virtual coordinates, and we set-up two matrices: a static matrix and a dynamic matrix.

The ID given to the robot corresponds to its radio turret ID, the radio base station has ID=0. *The virtual coordinates* assigned randomly at the beginning of the experiments and updated semi-randomly every 500 ms when robots are moving, for both X and Y coordinates a random value (-1, 0 or +1) is chosen and added to the old values of the virtual coordinates (Figure 3).

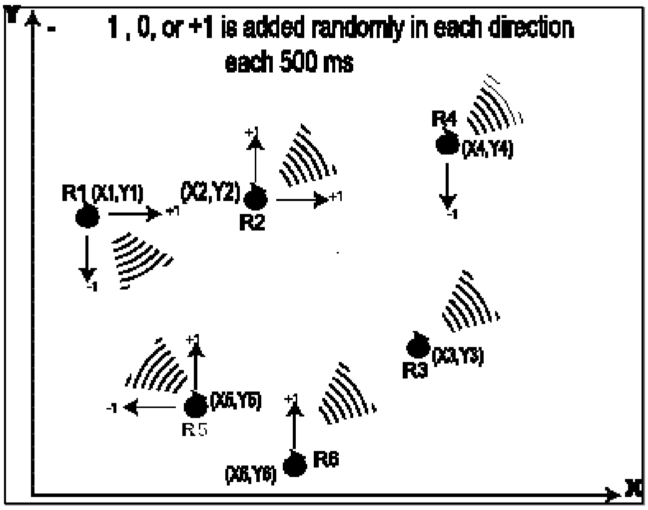

Figure 3. Semi-random update of the virtual coordinates.

The static matrix gives the initial configuration of the network, it maps the hardware configuration of the system, and does not change over the execution time, except if there are some robots added to or removed from the system. For a set of N robots, the static matrix STAT is a two-dimensional square matrix of size N, it gives the communication permissions of the robots at the initial time, according to the following schema:

STAT[I,J] = (permission, shape, size)

- STAT [I, J] = (0, 0, 0): no communication is allowed between the robot I and the robot J i.e. their communication devices are incompatible, and they are not allowed to communicate over all the execution time.
- STAT [I, J] = (1, X, Y): the robot I can communicate with the robot J, (I can send a message to J), according to a shape X of size Y. The shapes and sizes of communication were used to simulate the heterogeneity of the devices. The shape gives the family of the device, and the size gives the scope of that device.

In our experiments three families of communication were implemented, corresponding to three different shapes:

- *Rectangle*: to simulate the communication between a Moderok and a vision turret...etc.
- *Circle*: to simulate the communication between two radio turrets…etc.
- *Sector*: to simulate the communication between the Moderok and the video turret…etc.

Our model allows also heterogeneity in the same family of devices, when different sizes of communication are used in the same family.

The dynamic matrix, for the same set of N robots, is also a two-dimensional square matrix of size N. It is first initialized with the permissions of communication from the static matrix, and as soon as the robots are moving it is updated according to the following schema:

```
{

If is-In-Shape( I, J ) then

            DYNAMIC[I,J] = 1;
Else
          DYNAMIC[I,J] = 0;
}
```

The function is-In-Shape updates the permissions of communication during all the lifetime of the system. It tests for each pair of robots, if the communication is still allowed according to the updated virtual coordinates, it can be summarized as follows:

```
is-In-Shape (I,J)
{
 Switch shape (I,J)
 {
 case circle    return is-In-circle(I,J);
 case rectangle return
                is-In-rectangle(I,J);
 case sector    return is-In-sector(I,J);
 default    return False;
 }
}
```

The functions is-In-circle, is-In-sector and is-In-rectangle, test if the updated virtual coordinates of the robot J are included in the circle, sector or rectangle centered in the new virtual coordinates of the robot I, respectively. Finally the updated version of the communication network is given by a superposition of the static and the dynamic matrices as shown in (Figure 4).

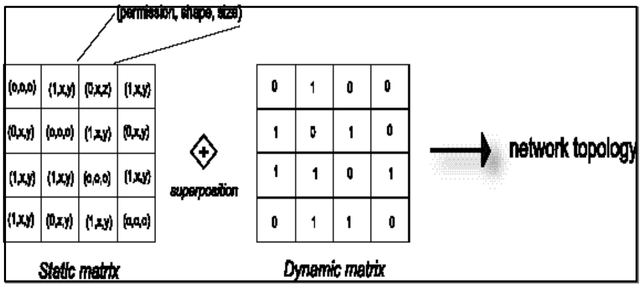

Figure 4. Superposition of the dynamic and the static matrices.

5. Experiments

We designed and implemented two sets of experiments: in the first set we used real Khepera robots, in the second set we used the Khepera simulator to quantify the performance of our system. It is worth emphasizing that in both sets of experiments, the robots operate completely autonomously and independently; all sensors, motors, controls are on-board, and there is no communication with the experimenters. The only possible interactions among robots are radio messages as a direct and dynamic form of interaction.

5.1. First Experiment

For this set of experiments we used five mobile Khepera robots endowed with the radio module, placed in a special arena. The arena we have used for the experiments consists of a copperplate by which the robots can collect the necessary current. This gives the robots an unlimited autonomy. The arena is about one square meter surrounded by white walls of about 20 cm height. Obstacles in our environment have a cylindrical shape with a diameter of around 2 cm and a height of 10 cm. We structured the environment in a way that limitations of the sensors of our robots are compensated.

Each robot runs three processes in parallel

a. *Alive process:* toggles the LED 0 every 500 ms, this process is used to test if the robot is alive, since it collects the necessary current from the copperplate, there could be some connection problems.

b. *Behavioral process:* implements the following individual behavioral rules of the robots.

- *Roam:* allows the Khepera to wander around in the arena, it implements a Braitenberg-based neural network that couples the values of the IR sensors with the speed of the wheels [9]. The IR sensors have an activation proportional to the proximity of an obstacle (Figure 6).
- *Avoid:* allows the Khepera to avoid obstacles deem to be in it's path, if an obstacle is present in the left hand side of the Khepera it moves toward it's right hand side, and conversely.
- *Go-Ahead:* used to make the Khepera go straight ahead.
- *Stop:* sets the robot motors speed to zero.

c. *Communication Process:* takes care of the communication between the robots, it implements the three following rules:

- *Send:* to send a message to a given radio turret or to the radio base station.
- *Rec:* to receive a message form a given Khepera or from the base station.
- *Broadcast:* used to broadcast a radio message to more than one Khepera, using the Broadcasting robot-based mode [7].

In our experiments the messages use the standard ASCII protocol of the Khepera.

We start our experiment by letting the five Kheperas wandering around, and performing obstacle avoidance in the arena, i.e. running the behavioral rules of the first state, after few minutes a special message is sent by the radio base station[5] to one of the robots chosen randomly. This special message is used in order to simulate an external event, like for instance a vocal alarm call that informs of a certain threat. When the robot receives the message it:

- switches its controllers to the second set of the behavioral rules i.e. moves straight a head and stops when the first obstacle[6] is encountered.
- propagates the message to its mates, to do so the robot asks first the radio base station for an up-to-date version of the network topology. This information is used to get the set of IDs corresponding to the robots with whom communication is allowed at that time, and broadcasts[7] the message to those robots.

In the same way, when receiving the message each robot performs the two previously cited operations: switch and propagate.

After running the experiment several times, our observations was the same for all the experiments, after a certain time, teams of different sizes have been formed in some regions in the arena. Thus we can conclude that the collective behavior has committed *"Agglomerations have been formed in the environment"*.

Our system exhibits the emergence: robots participate in changing and adapting the collective behavior of the system without being aware of it, but

[5] Using the global communication path.

[6] A robot is not differentiated from an object.

[7] Using the local communication path.

just in virtue of the design of their internal structure, which is based on operational autonomy, and using a dynamic form of communication. Moreover there is nowhere in the robots something encoded that specifies where and how to regroup themselves. The location and size of the agglomerations at the end of the experiment are not determined in advance, they are the result of the robot interactions and communications.

5.2. Second Experiment

Building from scratch a set-up for experiments with real robots can sometimes be hazardous. In most cases a simulation is indeed very useful, because it helps understanding several features within the system that would be prohibitive or even impossible to detect in a real configuration. It is precisely with this concern that we develop our simulation. Stressing on the measurement of quantitative results and on the realization of appropriate visualization tools to follow in real time the run of the system. For our simulations we used Khepera simulator version 2.0. The simulator gives a relatively faithful representation of the real Khepera by incorporating imprecise movements of the robot wheels, and introducing noise in the sensors measurements. Each robot is provided with 8 infra-red (IR) sensors used to detect other robots, obstacles and the arena walls[8].

We realized an intensive number of simulations of the experiment under the following conditions: there were 15 obstacles randomly distributed in the environment, and the number of robots ranged from 5 to 20. Figure 6 shows a snapshot of the simulation with 9 robots. The snapshot was taken when a system was in an intermediate state, the agglomerations have just start to be formed, and there are some robots that still are not aware of the external message.

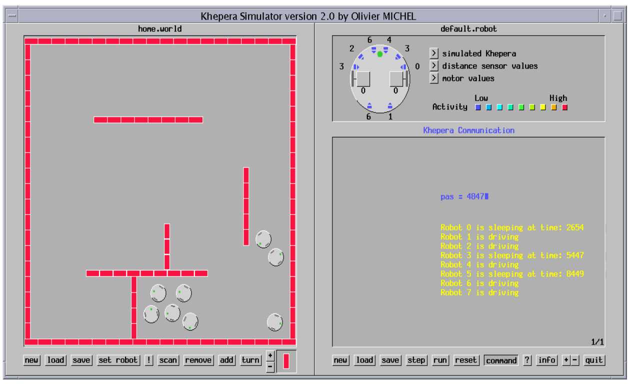

Figure 6. Snapshot of the simulation with 9 robots, in an intermediate state.

As we focus on the capability of the robots to adapt the collective behavior, a relevant feature for each experiment is the speed of adaptation i.e. the number of iterations necessary for the collectivity before the agglomerations start to appear in the arena. When performing our simulations we measured the speed-up of the system. Thereby we noted that to some extend, the more the number of robots in the system, the better the performance of the system is, and the much more faster the system converges to the desired behavior.

Therefore we come to the same conclusions as in the first set of experiments on the real Kheperas, that is, emergence of the collective behavior through simple behavioral rules and using a dynamic form of communication.

6. Summary and Future Work

In this paper we presented a collective robotics framework, where a team of heterogeneous communicating mobile robots, operating without a supervisor and without a centralized control of their behaviors, adapt the collective behavior during runtime in face of a changing environment. Our results show that simplistic behavior rules implemented in a decentralized manner can lead to complex collective behaviors. The two main advantages of our approach are:

- *Robustness and fault tolerance:* robots can fail, or be removed from the collectivity to some extent without affecting the system.
- *Scalability:* our architecture allows a dynamic integration of robots. More robots can be added easily and immediately participate in the system, without writing new functionalities, as long as the connectivity in the communication network is maintained.

In the current formulation, the communication devices have to form a communication network over the execution time, because if a subset of robots wanders out of range, they would be isolated from the rest of the team. This could also be considered as a feature rather than a problem, because sometimes one might want to divide the team to solve tasks in different parts of the environment. However robots that have to cooperate to solve a task have to form a network most of the time.

Till now we focused only on operational autonomy a next stage in the work will consist in studying behavioral autonomy, through learning approaches, in order to increase the adaptivity of the system. Another advantage will be possibly to make the collectivity reacts differently to different kind of events, just like some animals use a set of calls to signal different types of threats.

In conclusion, it should be mentioned that this was preliminary work so there are some problems that should be the focus of further investigations, but it shows promise in the search of collective adaptation of MAS and multi-robot systems.

[8] All sensors used in the simulations exist for the real Khepera robots.

References

[1] Chantemargue F., Lerena P., and Courant M., “Conflicts in a simple autonomy-based multi agent system,” *in Proceedings of AAAI'99, Workshop on agents conflicts*, Orlando, Florida, USA, July 1999.

[2] Chantemargue F., Lerena P., and Courant M., “Autonomy-based multi agent system: statistical issues,” *in Proceedings of Third World Multi-conference of Systemics, Cybernetics and Informatics (SCI'99) Fifth International Conference of Information Systems Analysis and Synthesis (ISAS'99)*, Orlando, Florida, USA, July 1999.

[3] Durfee E. H. and Lesser V., “Negotiating task decomposition and allocation using partial global planning,” *in Gasser L. and Huhns M.* (Eds.), *Distributed Artificial Intelligence*, vol. 2.

[4] Floreano D. and Mondada F., “Evolution of Plastic Neurocontrollers for Situated Agents,” *Maes P., Mataric M., Meyer J. A., Pollack J., and Wilson S. (Eds.), From Animals to Animats IV*, MIT Press, 1996.

[5] Jennings N. R., Sycara K. P., and Wooldridge M., “A Roadmap of Agent Research and Development,” *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 1, no. 1, pp. 7-36, 1998.

[6] Jennings N. R. and Wooldridge M., “Applications of Agent Technology,” *Agent Technology: Foundations, Applications, and Markets*, Springer-Verlag, 1998.

[7] K-Team SA, *Khepera Radio Turret User Manual*, Version 1, Lausanne, Switzerland, November 1999.

[8] K-Team SA, *Khepera K213 Vision Turret User Manual*, Version 1.0, November 1995.

[9] K-Team SA, *Khepera User Manual*, Version 4.06, Lausanne, Switzerland, November 1995.

[10] Minsky M., *Society of Mind*, Simon and Schuster, New York, 1985.

[11] Mondada F., Franzi E., and Inne P., “Mobile Robot Miniaturization: A tool for investigation in control algorithms,” *in Proceedings of IS-ER93*, Kyoto, October 1993.

[12] Pfeifer R. and Scheier C., *Understanding intelligence*, MIT Press, Cambridge, Massachusetts, 1999.

[13] Robert A., “EMuds: Adaptation in Text-Based Virtual Worlds,” *PhD Dissertation*, University of Fribourg, Switzerland, 1999.

[14] Uribe A. P., “Using a time-delay actor-critic neural architecture with dopamine-like reinforcement signal for learning in autonomous robots,” *Emergent Neural Comp. Architectures Based on Neuroscience*, LNAI 2036, Springer, pp. 522-533, 2001.

[15] Uribe A. P. and Courant M., “A robotics framework for studying the co evolution of signaling,” *in Proceedings of ICCS'2001*, Beijing, China, 2001.

[16] Uribe A. P. and Courant M., “Learning to predict variable-delay rewards and its role in autonomous developmental robotics,” *6th Int. Work-conference IWANN'2001*, LNCS 2085, Springer, part 2, pp. 492-499, 2001.

[17] Wooldridge M., “Agent-based computing,” *in Interoperable Communication Networks, Baltzer Journals*, vol. 1, no. 1, pp. 71-97, January 1998.

[18] Wooldridge M. and Jennings N. R., “Intelligent Agents: Theory and Practice,” *Knowledge Engineering Review*, vol. 10, no. 2, 1995.

[19] Wooldridge M., Mueller J. P., and Tambe M, “Intelligent Agents,” *Lecture Notes in Artificial Intelligence*, vol. 1037, Springer-Verlag, January 1996.

Soraya Mostéfaoui is a PhD candidate in the Parallelism and Artificial Intelligence research group (PAI) at the Computer Science Department at the University of Fribourg in Switzerland. Her research interests include distributed computing on the web, ubiquitous computing, collective robotics, multi-agent simulations, and emergent behaviors. Soraya received an engineer degree in computer science from the National Institute of Computer Science, Algiers, Algeria in 1999, and a master degree in computer science from the Swiss Federal Institute of Technology, Lausanne Switzerland in 2000. She is a member of the ACM.