Adaptive Patterns for Intelligent Distributed Systems: A Swarm Robotics Case Study

Mariachiara Puviani, Giacomo Cabri, and Letizia Leonardi

Abstract. In order to propose some evaluation of adaptive architectural patterns for intelligent distributed systems, in this paper we present a case study where a swarm of robots is required to coordinate and adapt to perform a task. We will present the results of several simulations and propose some considerations that help further research about adaptive patterns in intelligent distributed systems.

1 Introduction

Nowadays adaptation in intelligent distributed systems, that is defined as the ability of a system to change its behaviour to dynamic operating conditions [4], is a very important aspect for these systems. Adaptation can be applied to intelligent distributed systems at two levels: at the level of a *single component* (components are able to automatically change their behaviour according to the changes in their context) and at the level of a *whole system* (the entire system can exhibit adaptive capabilities). In the latter case adaptation can be achieved in two different ways: (i) by *explicitly* programming each component to manage a specific portion of the global goal; (ii) by achieving a global goal through a new behaviour that the whole system will exhibit even if composed of autonomous components. The latter described adaptation is what happens in *swarm robotics*: each robot acts individually using local information. The global goal of the system is achieved by means of the collective behaviour of all the components involved in the system.

Mariachiara Puviani
DISMI - Università di Modena e Reggio Emilia, Via Amendola 2, Reggio Emilia, Italy
e-mail: `mariachiara.puviani@unimore.it`

Giacomo Cabri · Letizia Leonardi
DII - Università di Modena e Reggio Emilia, Via Vignolese 905, Modena, Italy
e-mail: `{giacomo.cabri,letizia.leonardi}@unimore.it`

G. Fortino et al. (Eds.): Intelligent Distributed Computing VI, SCI 446, pp. 241–246.
springerlink.com

In this paper, we present swarm robotics' simulations to show how adaptivity comes out in different situations, and to study which adaptive pattern can be useful in different situations. This evaluation helps us to understand how different patterns work and to decide which pattern is more appropriate.

2 Swarm Robotics and Adaptive Architectural Patterns

Swarm intelligence scenarios are an ideal starting point to understand how adaptation works in intelligent distributed systems. In swarm robotics, *task allocation* [2] is a well known problem, where the *goal* of each robot is to search for food items placed in an area, and bring them back to the nest. Each robot has also the *sub-goal* of avoiding obstacles (e.g., walls, other robots or objects) on its way. Thus, each robot needs to change its route in order to avoid obstacles while adapting its behaviour to a diminishing number of available food items. The goal of the system is to increase the nest energy with food items. This energy tends to decrease during the simulation, due to the energy consumption of each robot. The main *constraint* of each robot is to avoid running out of batteries. Running outside the nest, robots are consuming energy taken from the nest itself. Each robot must save its energy in order to keep the whole system up and running.

In this context, the adaptive behaviour of the system can be defined by a pattern. An *adaptive architectural pattern*[1] is a conceptual scheme that describes a specific adaptation mechanism. It specifies how the component/system architecture can express adaptivity. When developing an intelligent distributed system that needs to be adaptive, such as a swarm robotics one, the use of an appropriate pattern that will enact adaptivity will help developers in their work. The pattern permits to the developer to be guided to make the system exhibit a required behaviour, even when unexpected situations occur. Unfortunately, it is not always possible to immediately recognize which is the most appropriate pattern that the developers can use when building intelligent distributed systems. Thus, it is useful to have guidelines that explain the features of each patten, so the developers can choose the one they think is the most appropriate one for their needs. In order to verify if a pattern is the appropriate one for a specific system, we use a "black-box" approach: we implement the system considering when it needs to adapt and under which conditions it will adapt. In this way we can consider, for each case study, the tolerance under which the system can adapt itself during its execution in order to continue to satisfy its goals and constraints. This tolerance describes an area where the system should remain to be considered adaptive.

In [1], we proposed a preliminary list of patterns and offered examples for their use. In this paper, we proceed to understand whether exploiting a specific pattern can be useful to implement an intelligent distributed system. In our swarm robotics case study we apply the pattern that in [1] we called "pattern based on swarm intelligence connected with the environment". This pattern addresses the problem of coordinating a large number of simple components (with limited knowledge) in order

---

[1] In the following, we use the term *pattern* to mean always *adaptive architectural pattern*.

to achieve a global goal, whose explicit representation is not possible. The collective behaviour results from components' behaviour adjusted by local environment conditions. The single components are not able to directly communicate one with the other, but an implicit communication is made using the *environment* that propagates the adaptation. So the environment plays the role of a strong stimulus for components that aims at modifying their internal dynamical behaviour indirectly.

3 Swarm Robotics Simulations

In order to simulate the robot behaviour for our case study, we use ARGoS[2] [3]. We define a simple arena where robots are free to move (see Figure 2). The robots must bring the food find in the arena, to the nest to increase the nest energy, and there they can also recharge their batteries, decreasing the nest energy.

The behaviour of single robots is probabilistically determined. Each exploring robot returns to the nest with probability *Pr* and starts exploring with probability *Pe*; these probabilities change depending on the situation, and this makes the behaviour of each robot adapt according to the behaviour of all the other robots and to environmental changes (i.e. change in the number of food items, addition of obstacles and so on). Every robot, while behaves in order to satisfy its goal, changes its probabilities. The other robots, sensing the changes that are propagated in the environment, adapt their behaviour in their turn. Doing that they change probabilities again, and make the system to continuously adapt. Each robot in our simulation uses the "pattern based on swarm intelligence connected with the environment" presented in Section 2. In the following, we show how adaptation of the ensemble of robots works under different operational conditions, and if the chosen pattern is always suitable for the studied system.

3.1 Changing Number of Food Items

In this simulation, we have a fixed number of robots (20) acting in the arena, and we set a variable number of food items (starting from 5 to 50 food items). We can observe the different behaviour of the system when we change the number of items: robots try to adapt their actions in order to avoid too long unfruitful explorations while trying to collect as much food as possible to increase the nest energy.

Figure 1 shows simulation results, changing the number of food items in the arena. Specifically, Figure 1 (a) shows the number of walking robots (WR) in time, while Figure 1 (b) shows the number of collected food items (CF) in time.

As expected, Figure 1 (a) shows that for the time from 0 to 750 sec, the behaviour of the system is quite the same for all the different scenarios: initially the robots are all out searching for food, and then they start to adapt to the environment. The robots stay out of the nest, looking for food, only if there is a large amount of food around, because their probability of finding food in a short time increases. In this situation,

---

[2] `http://iridia.ulb.ac.be/argos/`

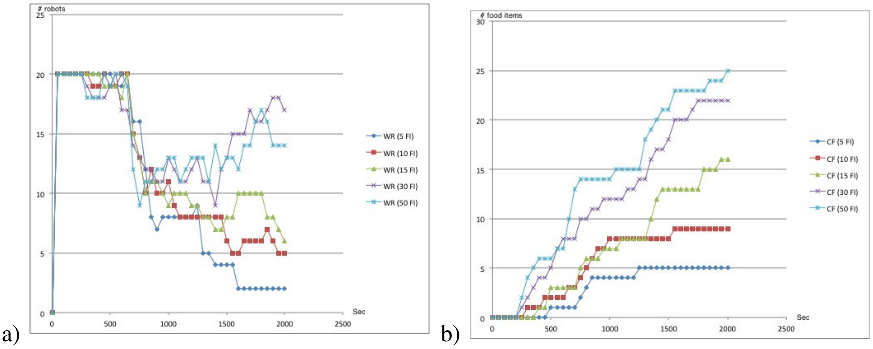

Fig. 1 Walking robots (a) and overall collected food items (b)

they are not losing energy and are motivated to resume exploring, after the pause in the nest, especially when the number of food items is higher than the number of robots. The chance of increasing the nest energy in this case is higher than the use of the energy itself. A robot can spend less than 50% of its energy to find the food and to return to the nest due to the large amount of food. Doing that, it increases the nest energy of more than what it spends. As an example, in this simulation the charged battery costs 1000 to the nest and each food item gives 500. When the food items present in the arena are more than 30, the average of battery consumption for each robot is 400, so there is a constant increase of energy (about 100).

We can see another adaptive behaviour in the system when the number of food items is low (5 Food Items - FI - or 10 FI): robots stay out of the nest for long time because they are more than the number of items in the arena, so their probability to return to the nest (*Pr*) and stay there rapidly increases.

Another expected result comes from the adaptation of the behaviour of every single robot: the number of collected items grows more rapidly when we have a higher availability of food in the arena (see Figure 1 (b)). That is because robots can quickly find food and their probability of staying out (*Pe*) for nothing decreases.

As a summarizing consideration, in this simulation the used pattern seems to be the appropriate one because the number of robots is fixed and the environment, that is the means of adaptation, is frequently changing, so it is the best way to propagate adaptation.

3.2 Changing Obstacles

In this simulation, we show how the system of robots adapts in environments with different obstacle locations. The number of robots is fixed (10), as the number of food items (15). Figure 2 shows the three experimental settings we used: the first one with no obstacles, the second with a short wall in the middle of the arena, and the third one with a long wall.

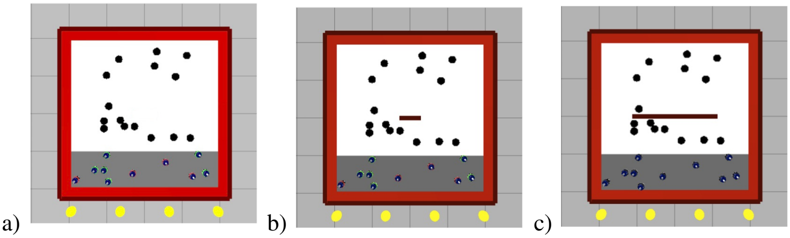

Fig. 2 Simulation arenas. White floor, grey nest and food items represented as black dots

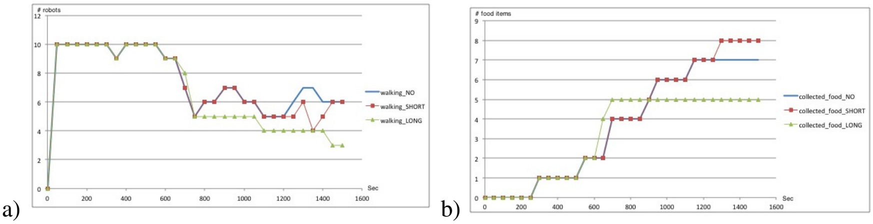

Fig. 3 Walking robots (a). Overall collected food items (b)

We can see from Figure 3 (a), that the number of walking robots is the same for the first part of the simulation (for around 700 sec). After that, the number of robots walking within the arena with a long obstacle sharply reduces. This happens because it is more difficult to find food and to come back to the nest, due to the obstacle. So robots spend more energy to avoid the obstacle and the probability to stay in the nest (*Pr*) is higher.

The same considerations can be done looking at the collected food (see Figure 3 (b)) in the three different settings. The number of food items that are brought to the nest is less when we have the long obstacle in the arena. It is interesting to observe that the number of items is larger (8 instead of 7) when there is the short obstacle in the arena, with respect to the case in which there is no obstacle at all. In fact, the short obstacle forced the robot to change their path to avoid it, and in this case, the change of path helps in finding the nest way or in finding a new food item.

In this simulation, we see that it can be useful to consider a different pattern than the used one, for the scenario with the long obstacle in the arena. Here a direct communication between robots is more useful to map the environment and help robots in their task: the obstacle makes robots that do not know the environment, frequently change their direction. This makes the robots to consume battery and not to be sure to find food items. An alternative pattern can be the one with a direct communication between robots, e.g. based on negotiation. The possibility to communicate how the external environment is in a given time tick, helps robots to find new food items and to localise the nest.

4 Conclusions

Using an appropriate pattern makes it possible to obtain an *intelligent* adaptive system even starting from components that behave simply in a probabilistic way and that have a limited knowledge about the environment and others components. The simulation reported in this paper suggest that some patterns are more suitable than others to build a specific system because they better specify adaptation mechanisms for the involved components and for the whole system. We shown that the "pattern of swarm intelligence connected with the environment" was the most appropriate for this kind of systems, for the majority of the cases. We have also shown that for some specific situations, this pattern is not the best one, and it would be better to apply another one that better suits the adaptive situation.

Our future work will focus on simulating others patterns to better understand their usage, and on enabling self-expression [5], which is defined as the capability of changing the whole pattern that descibes adaptation when the change of situation may require it (e.g. passing from the "pattern of swarm intelligence connected with the environment" to a pattern based on an "external adaptive manager" when the number of robots rapidly decreases, and the last pattern better manage the system).

Acknowledgements. Work supported by the ASCENS project (EU FP7-FET, Contract No. 257414).

References

1. Cabri, G., Puviani, M., Zambonelli, F.: Towards a taxonomy of adaptive agent-based collaboration patterns for autonomic service ensembles. In: 2011 Int. Conf. on Collaboration Technologies and Systems (CTS), pp. 508–515. IEEE (2011)
2. Krieger, M.J.B., Billeter, J.B.: The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. Robotics and Autonomous Systems 30(1), 65–84 (2000)
3. Pinciroli, C., Trianni, V., O´Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F., Stirling, T., Gutirrez, A., Gambardella, L.M., Dorigo, M.: ARGoS: a modular, multi-engine simulator for heterogeneous swarm robotics. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2011), pp. 5027–5034. IEEE Computer Society Press, Los Alamitos (2011)
4. Weyns, D., Holvoet, T.: An architectural strategy for self-adapting systems. In: Proc. of the Int. Workshop on Software Engineering for Adaptive and Self-Managing Systems, p. 3. IEEE Computer Society, Minnesota (2007)
5. Zambonelli, F., Bicocchi, N., Cabri, G., Leonardi, L., Puviani, M.: On self-adaptation, self-expression, and self-awareness in autonomic service component ensembles. In: Fifth IEEE Conf. on Self-Adaptive and Self-Organizing Systems Workshops (SASOW), pp. 108–113. IEEE (2011)