Bio-inspired Self-organization Methods and Models for Software Development

Daniel J. Dubois

Computer Science and Engineering
XXIII Cycle
`dubois@elet.polimi.it`

Introduction. The current research trends in Software Engineering are focusing on the development of new techniques to deal intelligently and efficiently with the design of systems that are able to evolve overtime and adapt to rapid changes of their requirements. In particular, the field of Autonomic Computing [1] has been created by IBM to study these types of systems with the ultimate aim to create systems that are able to self-configure, self-optimize, self-heal and self-protect without any external intervention. IBM proposed a layered architecture [2] composed of a controller layer and a controlled resource. The problem of this approach is that is does not scale well in systems composed of a high number of components such as pervasive systems [3], therefore in this research we propose an innovative decentralized solution that is based on the concept of self-organizing logic using analogies taken from the biological world to accomplish complex tasks using simple rules [4].

Self-organization is defined as *"The spontaneous evolution of a system into an organized form in the absence of external pressures"* [5]. The self-organizing logic we have cited above is composed of a specific type of algorithms (called self-organization algorithms) that are able to guide the evolution of the system toward a specific goal with no or minimal user intervention. An example of self-organizing system may be a vineyard humidity control system in which different parts of the terrain are equipped with humidity sensors that communicate to each other their humidity. One of these sensors is able to calculate the average humidity and to report problems if it goes beyond safety levels. It may happen that the battery of the aggregating sensor goes down, in such situation the system should be able to react and self-organize itself to elect a new aggregating sensor to keep the whole system in a working state. This is just a simple example of a traditional self-organizing system, however if we look at natural phenomena, we can see other more complex self-organizing systems, like ants finding the optimal path from the food source to the anthill [6], or fireflies that blink at the same time even if they are very far away in distance [7]. Forms of self-organization that come from observed phenomena taken from the natural world such as the ones discussed above are called *bio-inspired self-organization.*

The advantage of using algorithms inspired by natural bio-inspired systems with respect to more traditional approaches is the fact that they proved to be scalable (in terms of system size), robust (in terms of resistance to external perturbations), resistant to malicious of malfunctioning components (natural death of ants in an anthill), and so on [8]. However there is also a huge disadvantage:

we currently miss proper design approaches to support their application in real deployable systems. The final outcome of this research work is to study a proper methodology to address this issue.

Research Hypotheses and Directions. Our main focus in this research is to find some sort of *"ingredients"* for applying bio-inspired self-organization to a software system. We have identified three different main topics that will be discussed in the rest of this section: self-organization principles, self-organization algorithms, and development guidelines to apply them to real systems.

In previous work [9, 10] it has been found that different natural self-organizing system have some common patterns. We have extended this previous work by identifying some common principles that may be composed and translated into deployable algorithms. This way we have a sort of library of principles/algorithms along with the classes of problems they are able to solve.

The development guidelines we propose are the following: the first step is to analyze the problem to identify the correct principles and algorithms to use, then the second step is to build a model using either the algorithms as they are, or adopting some composed/reengineered version of them. This modeling step requires the model to be validated (in terms of convergence to the right solution) in some way. Previous work used either formal approaches such as *Temporal Logic*, *System Theory*, *Game Theory*, and *Operations Research*; as well as experimental approaches such as *Monte Carlo Simulations* and other statistical techniques. This is one of the most difficult tasks and it is definitely the point in which the work may take advantage from the material of other different research areas.

As soon as we are convinced that our model works, we can start implementing it. In this third step we should not be limited to a simple model-to-code approach since we have to face with non-trivial problems that might not be captured by the model such as components synchronization, race conditions, initialization and termination, frequency of algorithm iterations, amount of communication, and so on. Therefore the fourth and last step is to identify and solve real-systems problems using again either a formal or experimental approach. In conclusions the previous four steps are not necessarily sequential, but it should be possible to move backward/forward to the previous/next step depending on the necessity.

Current results we have already achieved include an application of bio-inspired principles where we managed to have a collaborative reduction of network traffic in a distributed-dispatcher publish-subscribe system [11].

Development and Evaluation Plan. The planned development direction of this work is to extend our current results to a more complete methodology consisting in a list of steps and design patterns that may be used to apply bio-inspired principles to real systems in a more systematic way with respect to the work that has been already done, especially in the validation steps. Other future developments include the following activities: propose new algorithms and improve existing ones, define problem classes and appropriate solutions, map problems/solutions to real case studies, and finally to compare the obtained results to traditional approaches.

Acknowledgements

I would like to thank my thesis advisor Elisabetta Di Nitto for her support in this thesis work and also Raffaela Mirandola for insightful discussions on the topic.

This research has been partially funded by the European Commission, Programme IDEAS-ERC (Project 227977-SMScom), FET (Project CASCADAS IST-027807), and FP7 (NoE S-Cube).

References

1. Kephart, J., Chess, D.: The vision of autonomic computing. Computer **36**(1) (2003) 41–50
2. IBM Corp.: IBM Autonomic Computing Toolkit - User's Guide. `http://download.boulder.ibm.com/ibmdl/pub/software/dw/autonomic/books/fpu3mst.pdf`
3. Waldrop, M.: Pervasive computing - an overview of the concept and exploration of the public policy implications (March 2003)
4. Saffre, F., Ghanea-Hercock, R.: Simple laws for complex networks. BT Technology Journal **21**(2) (2003) 112–119
5. Heylighen, F., Gershenson, C.: Information systems, may/june 2003. the meaning of self-organization in computing
6. Dorigo, M., Stützle, T.: Ant Colony Optimization. Bradford Book (2004)
7. Buck, J.: Synchronous Rhythmic Flashing of Fireflies. II. The Quarterly Review of Biology **63**(3) (1988) 265–289
8. Shackleton, M., Saffre, F., Tateson, R., Bonsma, E., Roadknight, C.: Autonomic computing for pervasive ict — a whole-system perspective. BT Technology Journal **22**(3) (2004) 191–199
9. Babaoglu, Ö., Canright, G., Deutsch, A., Caro, G.A.D., Ducatelle, F., Gambardella, L.M., Ganguly, N., Jelasity, M., Montemanni, R., Montresor, A., Urnes, T.: Design patterns from biology for distributed computing. ACM Trans. Auton. Adapt. Syst. **1**(1) (2006) 26–66
10. Kasinger, H., Bauer, B.: Design pattern for self-organizing emergent systems based on digital infochemicals. In: 2009 Sixth IEEE Conference and Workshops on Engineering of Autonomic and Autonomous Systems. (2009)
11. Dubois, D.J., Di Nitto, E., Mirandola, R.: Overlay self-organization for traffic reduction in multi-broker publish-subscribe systems. In: International Conference on Autonomic Computing '09. (2009)