

Self-expression in adaptive architectural patterns

Mariachiara Puviani

The self-expression mechanism enhances self-adaptation by enabling runtime selection of appropriate adaptive architectural patterns in response to environmental changes.

Engineering an autonomic system to be self-adaptive—that is, able to change its behaviour according to operating conditions encountered at runtime—helps simplify its design, and self-adaptive, self-aware components have been proposed to handle open-ended, highly parallel and massively distributed systems.¹ However, as complexity increases, self-adaptive systems become more difficult to develop, deploy and manage. Worse, what if the scenario changes at runtime to such an extent that the adaptive pattern that was most appropriate to the initial systems configuration is no longer relevant?

An adaptive architectural pattern is a conceptual scheme describing a specific adaptation mechanism for components and ensembles. The Autonomic Service-Component ENsembles (ASCENS) project was created to support the building of adaptive, self-aware autonomic systems by applying flexibility, low management overhead and optimal use of resources to traditional software engineering approaches.² Its research goal is to create and catalogue adaptive architectural patterns and present them along with usage guidelines explaining their features. Developers of self-adaptive components and systems will be able to select from these patterns or construct their own.³

A novelty of the ASCENS project is the introduction of the deeper concept of *self-expression*, by which a system may reshape its own configuration and organization dynamically whenever its patterns can no longer be effectively adapted. Self-adaptation allows a system to adjust its own operational parameters, while self-expression allows it to re-engineer itself entirely: the capabilities and architectures of its components, or even its own topology and control regime. Self-expression contextualizes self-awareness to specific adaptation needs, from ‘what I am and what is happening’ to ‘what I could become and how the world would change accordingly’.⁴

As a real-life example of self-expression, consider a scenario in which robots are deployed to locate and extract people trapped inside a partially collapsed building. One robot acting alone

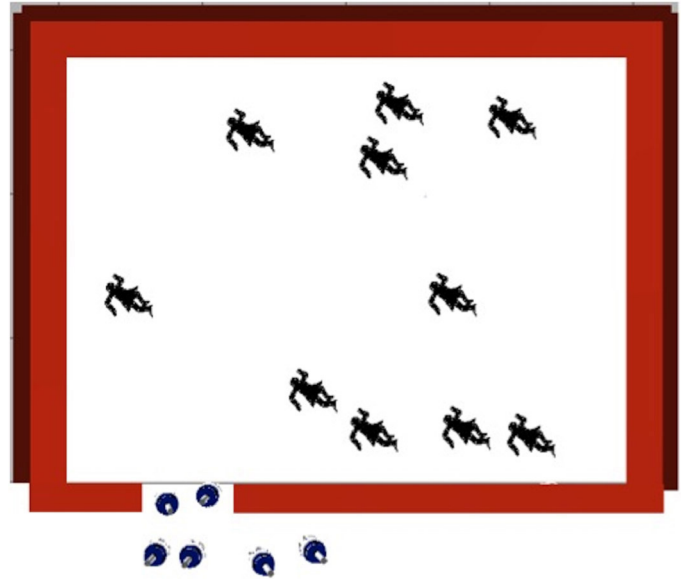


Figure 1. Swarm robots acting in a collapsed building. Simulation using the ARGoS system (Autonomous Robots Go Swarming).^{5,6}

might not be able to locate and remove every person quickly, even if it had a map of the building and knew how many people were inside. Multiple robots would have to contend with a highly dynamic and uncontrollable environment, due to the constantly shifting positions and states of people and of other robots. The system’s designer might take inspiration from the task allocation problem⁷ and choose the adaptive pattern based on swarm robotics shown in Figure 1, in which the group’s emergent behaviour results from the self-adaptive behaviour of individuals, influenced by local environmental conditions. This replaces the complexities of explicitly representing a global goal with the much more manageable deployment of simple entities that carry only limited knowledge of the environment and have little need to communicate directly with one another. The environment itself plays the role of a strong stimulus, providing implicit communication that indirectly modifies the internal dynamic behaviour of components and thus propagates adaptation.⁸

Continued on next page

Now suppose that the initial system requirements undergo a runtime contextual change, perhaps due to the introduction of objects not found on any map of the building (e.g. the fallen beams shown in Figure 2). A robot encountering one of these unanticipated obstacles loses substantial time in finding a path around it, and without a method of recording and relaying obstacle information to others, overall swarming performance decreases. One solution is to give the robots the ability to share the positions of obstacles with one another, but this would add complexity to their design and would compromise the autonomy that characterizes swarm robotics. Instead, the robots could be designed as self-expressing systems: initially not intercommunicating, they would only ‘re-engineer’ themselves to add that capability if it should become necessary.

Alternatively, a designer might choose to add self-expression not to the components (the robots) but to their environment. For example, suppose security cameras like the one shown at the corner of Figure 2, mounted throughout the building, are managed by an autonomic external system. Though this ‘centralized manager’ has no day-to-day need to detect extraordinary conditions such as fallen beams, self-expression would allow it to replace itself at the appropriate time with a system that can extrapolate the positions of newly present obstacles and communicate that information directly to any robots present. The advantage of this approach is that it applies adaptation logic to shared knowledge without adding complexity to the robots,

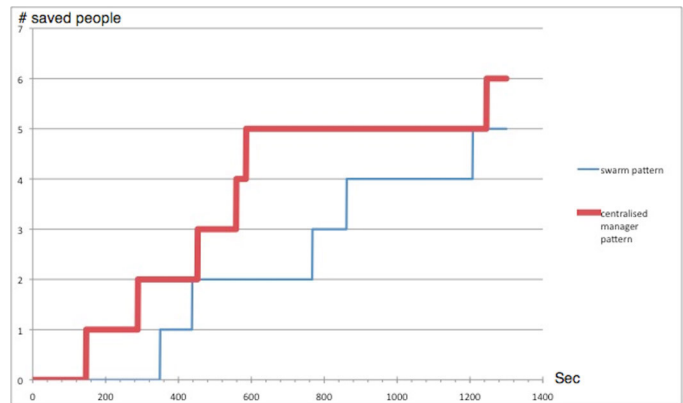


Figure 3. Number of people rescued over time, with and without a centralized manager to determine and communicate obstacle positions.

allowing all components to continue to act independently. The simulation results in Figure 3 show a clear improvement, and in fact performance in this case approaches that of the case with no obstacles.

Self-expression is a mechanism that enables components and ensembles to preserve effective adaptive capabilities, yet optimize overall performance, by re-engineering their structure dynamically according to multiple patterns. Our next step is to formalize self-expression mechanisms by defining methods by which systems may change from one adaptive pattern to another, thus creating a catalogue of patterns of self-expression.

This work was supported by the ASCENS project (EUFP7-FET, contract 257414).

Author Information

Mariachiara Puviani

DIEF (Enzo Ferrari Department of Engineering)
University of Modena and Reggio Emilia
Modena, Italy

Mariachiara Puviani earned her PhD in computer science from the University of Modena and Reggio Emilia in 2010 and has been a researcher in computer science there since 2011. She received her laurea in computer science engineering from the University of Ferrara in 2006.



Figure 2. Swarm robots acting in a collapsed building with fallen beams.

Continued on next page

References

1. D. Weyns and T. Holvoet, *An architectural strategy for self-adapting systems*, **Proc. 2007 Int'l Wkshp. Software Eng. Adapt. Self-Manag. Syst.**, p. 3, 2007.
2. <http://www.ascens-ist.eu> Autonomic Service-Component ENSEMBLES EU project. Accessed 4 October 2012.
3. G. Cabri, M. Puviani, and F. Zambonelli, *Towards a taxonomy of adaptive agent-based collaboration patterns for autonomic service ensembles*, **2011 Annu. Conf. Collaborative Technol. Syst.**, pp. 306–315, 2011.
4. F. Zambonelli, N. Bicochi, G. Cabri, L. Leonardi, and M. Puviani, *On self-adaptation, self-expression and self-awareness in autonomic service component ensembles*, **SASOW 2011: 5th IEEE Conf. Self-Adaptive Self-Organizing Syst. Wkshp.**, pp. 108–113, 2011.
5. <http://iridia.ulb.ac.be/argos> Autonomous Robots Go Swarming open source simulator. Accessed 4 October 2012.
6. C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, et al., *ARGoS: a modular, multi-engine simulator for heterogeneous swarm robotics*, **IROS 2011: Proc. IEEE/RSJ Int'l Conf. Intell. Robots Syst.**, pp. 5027–5034, 2011.
7. M. J. B. Krieger and J. B. Billeter, *The call of duty: self-organised task allocation in a population of up to twelve mobile robots*, **Robot. Auton. Syst.** 30 (1), pp. 65–84, 2000.
8. M. Puviani, G. Cabri, and L. Leonardi, *Adaptive patterns for intelligent distributed systems: a swarm robotics case study*, **IDC 2012: Proc. 6th Int'l Symp. Intell. Distrib. Comput.** In press.