

# Design and Modeling for Self-Organizing Autonomic Systems

Paul L. Snyder and Giuseppe Valetto

Drexel University, Philadelphia PA 19104, USA

**Abstract.** Describing, understanding, and modeling the behavior of systems built upon self-organizing principles (such as many bio-inspired systems) is key to engineering self-organizing systems that can solve problems in real computing environments. Capturing the properties of the micro-macro linkage that connects local behaviors of system components to global emergent properties of the system as a whole is particularly important. Different kinds of models have been proposed, each focusing on a different aspect of the problem: descriptive models provide notations that support the design activity and the application of self-organizing principles; validation models allow formal examination of dynamic properties; and analytic models provide techniques for mathematical exploration of abstracted collective behaviors. Our goal is to identify and select the best tools available from these families, extend them where needed, and tie them together to support the creation and analysis of self-organized autonomic computing systems in an integrated way.

## 1 Introduction

Modern computing environments continue to expand in scale and complexity, challenging traditional approaches to engineering and control. These systems place increasing burdens on the limited resources of skilled administrators, calling for the development of new approaches to design, maintenance and management. *Autonomic computing* is one such paradigm, with the goal of developing adaptive software that can to respond to changing conditions and environments without human intervention.

Some branches of autonomic computing have used *exogenous control* approaches drawn from fields such as control theory. More recently, there has been growing interest in *endogenous control*, where desirable global system behaviors emerge from the *self-organizing* interactions of many individual elements. This latter approach is of particular interest as computing systems reach ultra-large scales [1]. In the face of huge numbers of components with heterogeneous properties and ownership, high levels of dynamism, and indistinct system boundaries, centralized command and control approaches to autonomic problems (such as self-configuration, optimization, and diagnosis) become impractical.

Studies of naturally occurring self-organization in complex adaptive systems (particularly from biology) have proven to be a rich source of inspiration for building distributed computing systems along similar principles. Phenomena ob-

served in natural systems are often useful as high-level metaphors, but specifications with additional levels of formality and detail are needed to turn them into working distributed systems. Experience shows that bio-inspired metaphors often lead to computing systems with extreme *scalability* and *robustness*. Since their self-organizing rules operate in local neighborhoods, an individual computing element in a self-organizing system need only consider a small subset of the total information available in the system at any given time; as a consequence, systems designed using these principles may be able to scale more easily to very large numbers of interacting elements. These systems also show emergent adaptive behaviors, leading to redundancy and self-healing even when the system is perturbed or disrupted in new and unpredictable ways.

Self-organization and emergence are related concepts. Discussions of self-organization highlight the radically decentralized nature of an algorithm, which can achieve increased order (either temporal, spatial, or functional) without external direction. Emergence refers to a two-way linkage between the lower (*micro*) level interactions between individual system elements, and higher (*macro*) level properties exhibited by the collective system behavior, which are not explicitly specified in—nor reducible to—those lower-level interactions [2].

As the interplay between the micro- and macro-level can be elusive, there is an inherent difficulty in trying to predict and explain the behavior of a self-organizing system. Indeed, subjective surprise on the part of an observer has been proposed as a test for the existence of emergence [3].

Designers of self-organizing systems that must operate in real-world circumstances have to ensure that the such systems satisfy requirements across a wide range of operating conditions. As such, describing, understanding, and modeling the nature and mechanics of the micro-macro linkage in a system is key to engineering self-organizing systems, in order to develop some guarantees (for instance, statistical) on global-level properties, and to show that the system is robust and reliable across a range of conditions.

One way to reason about such systems is through expressive modeling techniques. There has been much recent research in this area, but it very much remains an open problem. Different models capture different concerns and try to explain different aspects of the problems of engineering self-organization and emergence. Certain approaches provide descriptions that support the design of new systems. Others provide techniques to validate the self-organizing dynamics of a system and generate proofs about them. Others build mathematical models for the analysis of their behavior at some abstraction level.

Design, validation and analysis are all important concerns. However, it is currently quite difficult to connect models from these various families, and achieve a faithful and insightful representation of a self-organizing computing system. We maintain that a multi-perspective modeling process is needed, and our goal is to identify and select the best available modeling techniques from each of these families, and integrate them to better support the creation of new systems, as well as the analysis of existing ones.

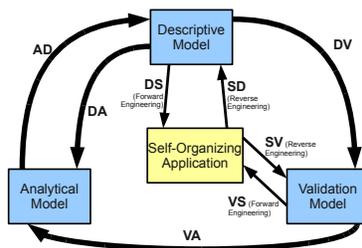


Fig. 1: Families of models for self-organized autonomic systems. Arrows represent feasible transitions, as discussed in Section 3

## 2 Motivating Scenario

The need for a set of tools allowing reusable design, validation, and analysis has been highlighted by our previous experiences building self-organizing systems. Several of these systems have centered around *Myconet*, an unstructured overlay protocol for peer-to-peer networks [4]. The Myconet algorithm takes inspiration by fungal growth patterns, in order to build an efficient self-optimizing superpeer topology that can also rapidly self-heal in response to damage or attacks.

Myconet has proven to be flexible, and we have used it as a platform for other self-organizing applications, including load-balancing in distributed service networks [5], and attack detection and mitigation [6]. Those works have demonstrated the applicability of Myconet to a variety of self-organizing scenarios and applications. New applications can be added as a layer on top of the core Myconet; to leverage some of its properties, they may need to interact with the core and may induce alterations to the lower-level behaviors.

Each extension to Myconet brings additional insights into self-organizing dynamics, but also show us the limitations of ad hoc approaches to design and analysis of each new application. These experiences have led us to investigate formal tools and models. We have examined a number of models and tools from the literature. Our intent is to select chosen techniques to model Myconet and its extensions, leading to the development of an integrated set of useful tools and modeling approaches that can be reused for the principled design of other Myconet extensions, and at the same time enhance our understanding of self-organizing design in general.

## 3 Proposed Research

The design of self-organizing control mechanisms requires tools beyond those offered by traditional software engineering, and a large number of models have been proposed in the literature. These models have been developed separately for different purposes, and they may examine very different features of a system. We have classified the models we have examined into three general families: *descriptive models*, *validation models*, and *analytic models*. These model families—with some relationships among them—are depicted in Figure 1. Descriptive models are used to support the design of self-organizing systems. They draw on and extend tools both from the software engineering (such as UML diagrams and

agent-oriented SE) and complex adaptive systems fields. Examples include several efforts to catalog *design patterns* for endogenous control, and arrange those patterns into composable ways [7]. Other research efforts have proposed descriptive notations (often as extensions to the Unified Modeling Language [UML]), using tools such information flow analysis [8], causal loop diagrams [9], or explicit modeling of a “meso-layer” [10], to explicitly bring self-organizing and emergent mechanisms into the design process. Most notations are only semi-formal, and as such may not provide much support for more rigorous analyses of system properties. A major use of descriptive models is to facilitate the actual development of a self-organizing application through a process of forward engineering (transition DS in Figure 1). The reverse engineering of a design model from a working self-organizing system (transition SD) is also possible; however, automated reverse engineering tools are not necessarily effective at highlighting the design elements that are significant to self-organization. In particular, automated identification of self-organizing patterns is an open problem.

The goal of validation models is to study the properties of a system, and verify that it fulfills its requirements. This topic has been studied extensively in software engineering, but the very complex interactions in self-organizing applications (which also tend to be highly distributed and highly concurrent), and their emergent nature, pose a great challenge to traditional validation approaches. Additional tools are needed in order to formally reason about the dynamics of such systems. Examples of proposed validation techniques include the use of formal languages, *e.g.* event calculus [11] and stochastic  $\pi$ -calculus ( $S\pi$ ) [12]. Other approaches use established formalizations from software validation research and the analysis of biological systems, such as Stamatopoulou *et al.*’s proposed hybrid of Communicating X-Machines and Population P Systems [13]. Validation models can also serve as starting points for developing system (transition VS). Pitt *et al.* [11] take an event calculus description of an observed sociological system and compile it into an executable simulation that exhibits actual self-organizing dynamics. Direct derivation of a validation model from an actual system by reverse engineering (transition SV) may also be possible.

Analytical models offer another approach to understanding self-organizing systems by reducing their dynamics to a set of simplified mathematical equations that often abstract away lower-level details. In the best case, an analytic model allows terse descriptions of global behaviors and provides a foothold for the exploration of an application’s parameter space. A number of analytical models use concepts from physics or information theory, and focus on characterizing the statistical complexity of the system, for example by measuring its entropy [14]. These approaches provide tools for assessing the self-organizing properties of a system once it has been developed, but are less helpful for initial system design.

In general, it is difficult to determine which model may be best for a particular system. Each modeling approach evidently has its own strong suit, but also carries some limitations. In some cases, is possible to translate between specific models belonging to different families in order to gain the benefits of each. For instance, it may be possible to produce a validation model starting from a de-

scriptive one; for example, Sudeikat *et al.* [12] offer a procedure for deriving a ( $S\pi$ ) model from a descriptive causal loop diagram (transition DV in the figure). Renz *et al.*, [9] propose a method for moving from a validation model (specified in  $S\pi$  calculus) to an analytic model in the form of a system of differential rate equations (transition VA). Transition in the opposite direction is hardly possible as the higher level of abstraction of analytic models results in loss of formal detail about the system. Moving directly from a descriptive to an analytical model may be possible, although the literature only suggests preliminary steps. For example, the specifications of self-organization design patterns may be decorated with analytic annotations about its behavior, as shown in [7]. Direct generation of a descriptive model from an analytic model (transition AD) is difficult, for the same reasons as transition AV: analytical models tend to abstract out a lot of detail. Since analytical models of biological phenomena (or other Complex Adaptive Systems) developed in the natural sciences are often used as the starting points for the design of new computing systems, it is typical to try to capture them by hand as thoroughly as possible within a suitable descriptive model.

While some works that demonstrate transitions between specific models in different families exist, they represent point solutions, and this problem is in general currently outstanding. Some recent research has also focused on how to move beyond an ad hoc approach to the design of self-organizing systems, and specify a methodology. Several design methodologies have been proposed, typically beginning with a descriptive model of a proposed system (*e.g.*, [15]); however, these methodologies do not directly address the roles of models from different families in their design process.

The issue of a more unified and comprehensive approach to the specification of self-organized systems is therefore an open area of investigation. The objective of our research is to understand how existing modeling techniques can be connected and integrated, when necessary by providing some extensions and bridges among them, so that we can solve in an integrated way the problems of sound construction, production of proofs and guarantees, and dynamics analysis, in the challenging domain of self-organized autonomic software systems. Such integration would not only represent an enhancement that may facilitate the work of researchers and practitioners involved in the development of those systems; it is also likely to lead to new insights on how different concerns interact and intertwine, and impact the creation of emergent mechanisms and self-organized dynamics, and analyze their efficacy.

## 4 Research Plan and Conclusions

All three types of models discussed in the previous section play an useful role in the engineering of self-organized autonomic systems, and research is needed to develop a unified approach to modeling.

In the initial stages of our research, we will select best-of-breed modeling techniques from each of the three families. We will apply these to our core Myconet platform, in order to understand the benefits and limitations of each, and extend them where necessary. Based on our experiences, we will develop a preliminary integration of the selected models, resulting in a collection of

transformations for moving between different models and a process for applying these. Next, we will extend the models to include the layered applications that have been developed on top of Myconet, and examine how they can be expressed as specializations of or extensions to the base Myconet models, while continuing to refine the integration. Finally, we intend to release Myconet and its models, along with the integrated collection of tools.

This research will result in insights into which of the large number of models proposed in the research literature are most useful and beneficial for designing and analyzing self-organizing systems in a unified way, as well as an integrated approach to applying those multiple modeling techniques. It will also provide a demonstration of a principled approach to the design of self-organizing applications using the Myconet platform, which will be instrumental in enabling its wider adoption. This unified modeling of Myconet may also serve as a blueprint or template of how the design and analysis of self-organizing applications in general can be addressed.

## References

1. L. Northrop, P. Feiler, R.P. Gabriel, J. Goodenough, R. Linger, T. Longstaff, R. Kazman, M. Klein, D. Schmidt, K. Sullivan, et al. Ultra-large-scale systems: The software challenge of the future. *Software Engineering Institute*, 2006.
2. Tom De Wolf and Tom Holvoet. *Emergence Versus Self-organisation: Different Concepts but Promising When Combined*. Lecture Notes in Computer Science. Springer-Verlag, 2005.
3. E.M.A. Ronald, M. Sipper, and M.S. Capcarrère. Design, observation, surprise! A test of emergence. *Artificial Life*, 5(3):225–239, 1999.
4. P.L. Snyder, R. Greenstadt, and G. Valetto. Myconet: A fungi-inspired model for superpeer-based peer-to-peer overlay topologies. In *SASO'09*, pages 40–50, 2009.
5. G. Valetto, P.L. Snyder, D.J. Dubuois, E. Di Nitto, and N.M. Calcavecchia. A self-organized load-balancing algorithm for overlay-based decentralized service networks. In *SASO'11*, 2011.
6. P. Snyder, Y. Osmanlioglu, and G. Valetto. Biologically inspired attack detection in superpeer-based p2p overlay networks. In *Proc. of BIONETICS*, 2011.
7. J.L. Fernandez-Marquez, J.L. Arcos, G. Di Marzo Serugendo, and M. Casadei. Description and composition of bio-inspired design patterns: the gossip case. In *Engineering of Autonomic and Autonomous Systems (EASE), 2011 8th IEEE International Conference and Workshops on*, pages 87–96. IEEE, 2011.
8. T. De Wolf and T. Holvoet. Using uml 2 activity diagrams to design information flows and feedback-loops in self-organising emergent systems. In *Proceedings of the Second International Workshop on Engineering Emergence in Decentralised Autonomic Systems (EEDAS 2007)*, pages 52–61, 2007.
9. W. Renz and J. Sudeikat. Modeling feedback within mas: A systemic approach to organizational dynamics. *Organized Adaption in Multi-Agent Systems*, pages 72–89, 2009.
10. M.A. de Cerqueira Gatti and C.J.P. de Lucena. A bio-inspired representation model for engineering self-organizing emergent systems.
11. J. Pitt, J. Schaumeier, and A. Artikis. The axiomatisation of socio-economic principles for self-organising systems. In *Proc. of IEEE Conf. on Self-Adaptive and Self-Organizing Systems (SASO2011)*, 2011.
12. J. Sudeikat and W. Renz. A systemic approach to the validation of self-organizing dynamics within mas. *Agent-Oriented Software Engineering IX*, pages 31–45, 2009.
13. I. Stamatopoulou, P. Kefalas, and M. Gheorghe. Modelling the dynamic structure of biological state-based systems. *BioSystems*, 87(2-3):142–149, 2007.
14. D.C. Marinescu, J.P. Morrison, C. Yu, C. Norvik, and H.J. Siegel. A self-organization model for complex computing and communication systems. In *Self-Adaptive and Self-Organizing Systems, 2008. SASO'08. Second IEEE International Conference on*, pages 149–158. IEEE, 2008.
15. Tom De Wolf and Tom Holvoet. Towards a methodology for engineering self-organising emergent systems. 2005.