

Modelling is key to engineering self-organizing software systems

Giuseppe Valetto and Paul L. Snyder

Designing self-organizing software requires an integrated modelling toolkit that encompasses design, validation and analysis.

We are witnessing, across application domains as disparate as ubiquitous and pervasive computing, peer-to-peer networks and computational grids and clouds, a trend toward a new generation of ultra-large-scale systems.¹ In such conditions, not only human-intensive management, but even centralized, command-and-control automated approaches to self-adaptation and self-management quickly become impractical. For these reasons, there is growing interest in endogenous control, where desirable global system behaviours emerge from the self-organizing interactions of many individual elements. Although research has demonstrated the usefulness and benefits of self-organized software systems, their systematic engineering remains an open problem. We investigate the critical need for models that can describe, analyse and validate self-organizing behaviours in software.

The design of software based on self-organizing principles requires tools beyond those offered by traditional software engineering. Prevalent techniques (Unified Modelling Language models, for example) are well equipped to represent the structure and internal dynamics of individual components, or the deterministic interactions between components. They are not as well suited, however, for capturing self-organizing systems, where local actions taken autonomously by large populations of components induce subtle feedback loops that lead to system-wide, emergent behaviours.

Many recent publications on the modelling of self-organization have tried to fill that gap, but they address individual facets of modelling with disparate techniques. A methodology that reconciles those facets and techniques, and moulds them into a coherent and systematic modelling toolkit for software engineers is missing.

Models can provide leverage on the difficult and important problem of connecting local, autonomous behaviours of

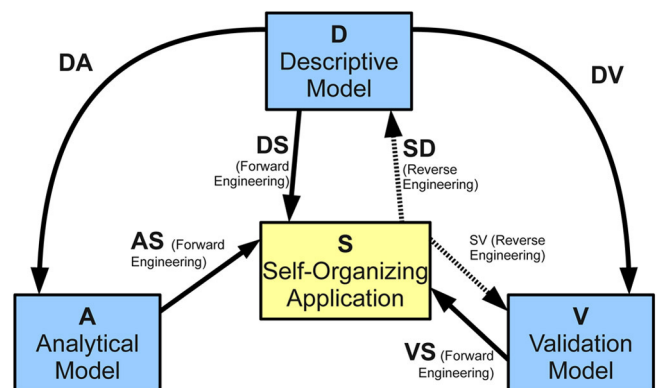


Figure 1. Families of models for self-organizing systems and possible transformations between them.

individual components to global, emergent properties of a system. The state of the art has tried to attack this problem from three different points of view, as discussed in our other work² (see Figure 1): *Descriptive models* assist in the design of new self-organizing software systems or in the modularizing of mechanisms implemented in existing systems. *Analytical models* (such as differential equations, similar to those frequently used in studies of self-organization in natural systems) support the analysis of system behaviours through mathematical abstractions. *Validation models* (including formal calculi) help to explore the space of system dynamics and make statements about what system-wide properties hold in different parts of that space.

The need for a set of tools that integrates multiple modelling perspectives has been highlighted by our previous experience, which has revolved around Myconet,³ a self-healing and self-optimizing, fungi-inspired peer-to-peer overlay protocol. Because Myconet has proven to be flexible, we have started to use it as a platform for building other self-organizing applications in large-scale distributed systems, such as load-balancing in decentralized service networks (Mycoload)⁴ and

Continued on next page

topology-based attack detection and mitigation using hormone diffusion (Hormone-Inspired Topology Adaptation Protection, or HITAP).⁵ Each new application is layered on the basic Myconet topology management protocol or on other applications. A modelling facility that supported designers in establishing and optimizing the desired properties of new applications, while ensuring that properties of other layers continued to hold, would be invaluable.

We have thus started to select a set of best-of-breed models from the literature in order to develop an integrated modelling suite covering design, validation and analysis. We want to apply the resulting suite to Myconet applications, and also propose it as a toolkit for engineering self-organized software in general.

Our starting point is the research on self-organization design patterns, which represents an example of descriptive modelling. We have undertaken an effort to reverse-engineer Myconet using a catalogue of composable self-organization patterns.⁶ That work has allowed us to isolate and highlight a number of distinct self-organization mechanisms.⁷ It has also enabled the identification and codification of two novel patterns: *Specialization* (a mechanism widely used in complex systems for achieving improved efficiency by exploiting the natural heterogeneity of system entities) and a generalized version of *Collective Sort* (a clustering mechanism that enables segregation or relocation of entities into similar-type groupings within the context of a collection of system elements).

We have also augmented the resulting pattern-based model with causal loop diagrams (CLDs), which are often used to describe and reason about complex adaptive systems, and have also been used by Sudeikat and Renz to represent self-organizing coordination mechanisms.⁸ CLDs are useful for making explicit the feedback loops that exist in self-organized systems, and which are essential to their functionality.

Although this resulting descriptive model is only semiformal, we consider it to be an apt foundation that will enable us to build transformations into both validation and analytical models. Validation notations can potentially be attached to, and enrich, either design patterns or individual transitions in feedback loops. Analytical formulas can be used to describe and formalize the emergent, system-wide behaviour generated by either an individual feedback loop or multiple interacting loops.

In summary, we have begun developing a suite of models for self-organizing systems that will combine the descriptive, analytical and validation approaches. Our descriptive model of the self-organizing protocol Myconet is based on the concept of self-organization design patterns. Our research will next focus on understanding what are the most suitable validation and

analytical modelling techniques, and the most appropriate 'hooking points' for integrating them with this kind of descriptive model. In the longer term, we are also assessing ways to include self-models from each of the families in the runtime code, in order to drive self-awareness and enable further adaptation.

Work on Mycoload is a collaboration with Elisabetta Di Nitto, Nicolò Calcavecchia and Daniel Dubois at Politecnico di Milano. Our work on self-organizing design patterns is a collaboration with Jose Fernandez-Marquez and Giovanna Di Marzo Serugendo at the University of Geneva.

Author Information

Giuseppe Valetto and Paul L. Snyder

Department of Computer Science
Drexel University
Philadelphia, PA, USA

Giuseppe (Peppo) Valetto joined the Department of Computer Science as an assistant professor in September 2007. He holds a Laurea in electronic engineering from the Politecnico di Torino (Italy) and an MS and PhD in computer science from Columbia University. His research interests include the engineering of autonomic systems, focusing on self-organization in large-scale software systems. He also studies cooperation and coordination in collaborative software development, in particular in large-scale, distributed software projects.

Paul Snyder is a PhD candidate. He holds a BA in computer science from Kalamazoo College in Michigan and an MS from Drexel University. His research interests are in self-organization and emergence in distributed software systems.

References

1. L. Northrop, P. Feiler, R. Gabriel, J. Goodenough, R. Linger, T. Longstaff, R. Kazman, *et al.*, **Ultra-large-scale systems: the software challenge of the future**, Software Engineering Institute, Pittsburgh, 2006.
2. P. Snyder and G. Valetto, *Design and modeling for self-organizing autonomic systems*, Tech. Rep. DU-CS-12-01, Drexel University, 2012.
3. P. Snyder, R. Greenstadt, and G. Valetto, *Myconet: a fungi-inspired model for superpeer-based peer-to-peer overlay topologies*, **Proc. SASO'09**, pp. 40–50, 2009.
4. G. Valetto, P. L. Snyder, D. J. Dubois, E. Di Nitto, and N. M. Calcavecchia, *A self-organized load-balancing algorithm for overlay-based decentralized service networks*, **Proc. SASO'11**, pp. 168–177, 2011.
5. P. Snyder, Y. Osmanlioglu, and G. Valetto, *Biologically inspired attack detection in superpeer-based p2p overlay networks*, **Proc. Bionetics 2011**, pp. 99–114, 2011.
6. J. L. Fernandez-Marquez, G. Di Marzo Serugendo, S. Montagna, M. Viroli, and J. L. Arcos, *Description and composition of bio-inspired design patterns: a complete overview*, **Natural Computing**, in press. doi:10.1007/s11047-012-9324-y

Continued on next page

7. P. Snyder, G. Valetto, J. L. Fernandez-Marquez, and G. D. M. Serugendo, *Describing self-organizing software with design patterns: a reverse engineering experience*, Tech. Rep. DU-CS-12-07, Drexel University, 2012.
8. J. Sudeikat and W. Renz, *Toward systemic MAS development: enforcing decentralized self-organization by composition and refinement of archetype dynamics*, **Proc. EEMMAS 2007**, pp. 39–57, 2008.