Discrete-Event System Performance Modeling of Self-Organizing Systems

Patrick Wüchner and Hermann de Meer
Chair of Computer Networks and Computer Communications,
Faculty of Computer Science and Mathematics,
University of Passau,
Innstr. 43, 94034 Passau, Germany
Email: {Patrick.Wuechner, Hermann.DeMeer}@Uni-Passau.de

***Abstract*— The contribution of this paper is twofold. On the one hand the authors give a brief survey on existing publications on the performance modeling of discrete-event systems and on the existing literature on the modeling of self-organizing systems. On the other hand an evaluation is started towards answering the question if modeling techniques designed for discrete-event systems are capable of describing self-organizing systems. We demonstrate that self-organizing systems share many properties of discrete-event systems. Thus, the well-known methods developed for the performance investigation of discrete-event system performance seem to be attractive for the performance evaluation of self-organizing systems.**

I. Introduction

Self-organization is supposed to play a decisive role in future communication systems, especially in the Internet. With the help of self-organizing systems, developers of already complex and still further growing communication infrastructures hope to keep their system manageable, scalable, configurable, repairable, and last but not least high-performing.

Thus, to see the benefit of self-organizing systems it is worthwhile to investigate modeling techniques suitable for evaluating the performance of these systems. Particularly with regard to complexity and nonlinearity, the modeling of self-organizing systems and the evaluation of these models is a non-trivial task.

This paper lists some ideas how to deal with this problem and discusses to which degree discrete-event system models are suitable to model the behavior of self-organizing systems. The investigation is consciously kept general and abstract to provide a common universal basis for multiple applications like sensor networks and ubiquitous computing, ad-hoc and group-forming networks, P2P networks, and others.

The paper is organized as follows. In Section II modeling techniques for discrete-event systems are reviewed. In Section III the paper introduces the properties of self-organizing systems. Section IV is used to discuss which properties complicate the modeling of self-organizing systems. Furthermore, Section IV investigates if self-organizing systems can be modeled and evaluated easily using the well-known methods developed for discrete-event systems.

II. Discrete-Event Systems

To set a basis for discussion, we will recapitulate the definition and different modeling and evaluation techniques for discrete-event systems (DESs).

A. Working Definition

Referring to a point of view which is widely accepted in large parts of the research community (see, e.g., [1]), a system in general can be described as a set of objects that depend on each other or interact with each other in some way to fulfill some task. While doing this, the system might be influenced by the system's environment. It is essential to set clear boundaries between the system and its environment.

At this point it is worthwhile to define some system components (see [1]):

- *Entity:* An entity is an object of interest in the system, whereby the selection of the object of interests depends on the purpose and level of abstraction of the study.
- *Attribute:* Attributes are used to describe the properties of the system's entities.
- *State:* The state of the system is a set of variables that is capable of characterizing the system at any time.

- *Event:* An event is an instantaneous incidence that might result in a state change. Events can be endogenous (generated by the system itself) or exogenous (induced by the system's environment).

Systems can be classified as continuous-state or as discrete-state systems (see [1]). In *continuous-state* systems the state variables change continuously over time. By contrast, the state of *discrete-state* systems only changes at arbitrary but instantaneous points in time.

An orthogonal classification of systems is based on the distinction if the progress of the dynamic system state is pushed forward by time or by the occurrence of events (see [2]). While in *time-driven* systems all state changes are synchronized by the system clock, in *event-driven* systems events occur asynchronously and possibly concurrently/simultaneously.

Thus, a DES (see, e.g., [1] or [2]) is a discrete-state, event-driven (not time-driven) system, i.e., the state changes of the system depend completely on the appearance of discrete events over time.

B. Modeling Techniques

Various approaches to the modeling of (discrete-event) dynamic systems exist (see [3]). These approaches can be divided into *logical models* and *timed models*. The latter are also known as *performance models* and can further be split into the classes of *deterministic* (or *non-stochastic*) and *stochastic* models.

Logical models and non-stochastic timed models are primarily used in control theory (see, e.g., [4]). These models mainly consider the logical aspects of the system, e.g., the order of events. Although it might be interesting to apply control theory to self-organizing systems, these models are not within the scope of this paper.

Thus, in this paper we focus on stochastic models of discrete-event systems. Event-driven systems may be modeled in *continuous time* or in *discrete time*. Several modeling techniques exist, developed especially for handling performance aspects of discrete-event systems, including *Markov chains*, *queueing networks*, and stochastic *Petri nets*. A thorough investigation of these well-established techniques can be found in [5].

C. Evaluation Techniques

Various evaluation techniques exist for the modeling techniques mentioned in Section II-B (see, e.g., [5]). These evaluation techniques include analytical/numerical methods (e.g., Markovian analysis, algorithms for product-form queueing networks) as well as (discrete-event) simulation.

Especially when dealing with very complex systems the following drawbacks can be observed:

- With Markovian analysis, it is necessary to create the whole state space (e.g., in form of generator matrices of continuous-time Markov chains). Due to the finite memory of computer systems, the calculation of results heavily suffers from state-space explosion. Furthermore, Markovian analysis can only be done for models that fulfill the Markovian property (i.e., state sojourn times have to be memoryless, i.e., have to be exponentially or geometrically distributed).
- Algorithms for product-form queueing networks require queueing network models in product form, which is a hard constraint. Furthermore, queueing networks lack means for handling synchronization and concurrency of system components.
- To get narrow confidence intervals it is necessary to simulate complex systems with long simulation runs and/or with a large number of simulation runs, especially in the presence of rare events. Thus, although simulation is a versatile tool because it can handle stochastic processes that fall into the class of generalized semi-Markov processes, it is a very time-consuming method for deriving results.

Of course, workarounds have been presented to these and further problems (see, e.g., [5], [6], or [7]). These include largeness avoidance, state space truncation, lumping, decomposition, fluid models, stiffness avoidance, stiffness tolerance, phase approximations, and simulation speed-up techniques.

Especially this plentifulness of methods makes it worthwhile to investigate if they can be used efficiently in the modeling process and evaluation of self-organizing systems.

III. Self-Organizing Systems

The term "self-organization" (SO) is widespread in science among various disciplines. Interdisciplinary lectures at the University of Passau have shown that it is difficult to find a common interpretation.

A. Working Definition

We utilize the view on self-organization that emerged within the USENET Newsgroup *comp.theory.self-org-sys* and was summarized in the "Self-Organizing Systems (SOS) FAQ"[1].

[1]Version 2.99, July 2006, http://www.calresco.org/sos/sosfaq.htm

There, SO is defined as follows:

- The evolution of a system into an organized form in the absence of external pressures.
- A move from a large region of state space to a persistent smaller one, under the control of the system itself. This smaller region of state space is called an attractor.
- The introduction of correlations (pattern) over time or space for previously independent variables operating under local rules.

The following typical features of SO are listed in rough order of generality:

- Absence of external control (autonomy)
- Dynamic operation (time evolution)
- Fluctuations (noise/searches through options)
- Symmetry breaking (loss of freedom/heterogeneity)
- Global order (emergence from local interactions)
- Dissipation (energy usage/far-from-equilibrium)
- Instability (self-reinforcing choices/nonlinearity)
- Multiple equilibria (many possible attractors)
- Criticality (threshold effects/phase changes)
- Redundancy (insensitivity to damage)
- Self-maintenance (repair/reproduction metabolisms)
- Adaptation (functionality/tracking of external variations)
- Complexity (multiple concurrent values or objectives)
- Hierarchies (multiple nested self-organized levels)

Similar views are (partly) shared by several researchers including, e.g., [8], [9], and [10].

B. Application Areas of Self-Organizing Systems

As stated before, "self-organization" was introduced into the vocabulary of researchers from many disciplines. Correspondingly, the field of application areas for self-organizing systems (SOSs) has grown quite large. For example, SOSs were related to: peer-to-peer systems (see [11] or [12]), social networks (see [13]), artificial neural networks (see [14]), and many more.

C. Performance Modeling Techniques

The approaches towards modeling self-organizing systems are mostly application area specific. They reach from very abstract, mathematical models based on Markov chains (like given in [15]) to application-oriented models, e.g., based on network calculus (see, e.g., [16]).

A widely preferred evaluation technique of SOS models is simulation, used, e.g., in [13] for studying social networks, in [17] for studying sensor networks, in [18] for studying neural networks, in [19] for studying mesh transport networks, in [20] for studying neuro mechanical networks, or in [21] for studying load balancing in grids. It is also interesting that although simulation is used a lot for evaluating SO models, the formal specifications of the underlying (discrete-event) system models are given rarely. Furthermore, no discussion could be found within these publications, why simulation was preferred to analytical solutions.

IV. Discrete-Event Self-Organizing Systems

To check if SOSs systems can be modeled appropriately by DESs, selected features of SOSs that were listed in Section III-A will now be discussed and related to DESs in more detail.

A. Autonomy

The autonomy of an SOS has to be distinguished from total autarchy. The latter does not exist in SOSs (see [9]). In contrast to total autarchy, interchanges (of, e.g., information, space, energy, or perturbations) between the SOS and its surrounding environment occur. In addition to these stimulations, the environment must not impose any control upon the SOS.

In terms of DESs, these stimulations can be modeled as exogenous events, e.g., in form of workload, job, token, or customer arrivals, as known from higher-level DES modeling formalisms like queueing networks or stochastic Petri nets.

B. Dynamics

As an event-driven system, a DES will have the facility to proceed as long as exogenous events occur. If no events are invoked by the environment, the DES's progress will depend on the existence of endogenous events. Regarding this, DESs show the same properties as SOSs.

C. Fluctuations

The term fluctuation may refer to two different aspects.

On the one hand, it describes fluctuation in the timing of events. This can quite easily be handled by stochastic DES performance models to some extend via defining stochastic distributions for inter-event times.

On the other hand, the means for modeling fluctuation of the SOS's structure is limited in DES models. One could discuss whether fluctuation in structure can be modeled by using routing probabilities as known from queueing networks or by using some construct of immediate transitions with marking dependent weights

(in the following abbreviated as MDWITs) in stochastic Petri nets to let the workload choose only entities (paths/transitions and nodes) that are set active for the current structure.

Usually, SOSs are too complex to know all possible shapes of the SOS's structure in advance. Thus, the approach will not be viable without vast abstractions.

D. Symmetry Breaking

Symmetry breaking is closely related to bifurcation. Bifurcation causes sudden qualitative or topological changes of the SOS's behavior. Analogously to fluctuations, bifurcations could — respecting the limitations — be modeled using concepts like MDWITs.

E. Local interactions

To ensure the locality of interactions, MDWITs could be defined in a way that they depend on the states of near-by entities only. For this, however, a metric for the term "near-by" has to be defined. Furthermore, it has to be discussed how this can be modeled dynamically and conveniently using high-level model descriptions.

F. Emergence

The investigation of emergence of the SOS towards a global order is the purpose of the study of SOS models. It is particularly interesting to investigate the properties of the attractors the SOS may move to.

Transient evaluation of short, mid, and long-term behavior could be observed using well-known analytical/numerical or simulation methods – presuming SOSs can be modeled as DESs.

G. Energy Usage

In the scope of DESs, energy usage could be modeled as stimulations from the environment that get dissipated by the SOSs. As already mentioned in Section IV-A, these stimulations can be modeled as exogenous events.

The dissipation of the energy can be modeled, e.g., by join constructs (known from fork-join systems described in [5]) that can be easily modeled using, e.g., stochastic Petri nets.

H. Nonlinearity

Nonlinearity results from feedback. Feedback can be positive or negative (see [9]). Positive and negative feedback could be modeled by fork and join constructs, respectively, eventually combined with state or marking-dependent transition rates and/or arc multiplicities.

Nevertheless, the nonlinear behavior of SOSs is usually more complex and this makes it hard to model SOSs and to evaluate these models. It is unlikely that this complex behavior can be (approximatively) described appropriately by linear systems of first order differential equations as it is done for DESs.

I. Further Features

The investigation of *multiple equilibria* is related to the investigation of the SOS's behavior (see IV-F).

Criticality (in particular *threshold effects*) could be modeled using MDWITs (see IV-C).

Redundancy and *repair mechanisms* could be handled by well-known methods taken from performability modeling (see, e.g., [6] or [22]).

Model *hierarchies* are also discussed in literature for several years (see, e.g., [5]).

V. Conclusion and Future Work

Although initial results for the investigation of correlations of DESs and SOSs were presented in this paper, the modeling of SOSs in an universal way and the evaluation of these models remains a non-trivial task.

Not all features of SOSs could be checked in detail yet whether they restrict or even prohibit the modeling and evaluation of SOSs with methods originally deployed for DESs. More time and research effort will be needed to complete a thorough investigation.

In near future, we plan to build a more detailed survey and classification of models used by researchers dealing with SOSs. From these models we might be able to learn how to extend the capabilities of DESs to increase their flexibility while modeling SOSs.

Particularly, we will have a look at modeling and evaluation techniques for nonlinear systems.

Acknowledgment

Parts of this work have been accomplished under support of the Euro-NGI – Network of Excellence, EC grant IST–507613.

References

[1] J. Banks, J. S. Carson II, B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation*, 3rd ed. Prentice-Hall, 2001.

[2] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, September 1999.

[3] P. J. G. Ramadge and W. M. Wonham, "The control of discrete event systems," in *Proceedings of the IEEE*, vol. 77, no. 1. IEEE Computer Society, jan 1989.

[4] J. G. Thistle, "Logical aspects of control of discrete event systems: A survey of tools and techniques," in *Proc. 11th International Conference on Analysis and Optimization of Systems – Discrete Event Systems*, ser. Lecture Notes in Control and Information Sciences, G. Cohen and J.-P. Quadrat, Eds., vol. 199. Berlin: Springer-Verlag, June 1994, pp. 3–15.

[5] G. Bolch, S. Greiner, H. de Meer, and K. Trivedi, *Queueing Networks and Markov Chains*, 2nd ed. New York: John Wiley & Sons, 2006.

[6] K. S. Trivedi, G. Ciardo, M. Malhotra, and S. Garg, "Dependability and performability analysis using stochastic Petri nets," in *Proc. 11th International Conference on Analysis and Optimization of Systems – Discrete Event Systems*, ser. Lecture Notes in Control and Information Sciences, G. Cohen and J.-P. Quadrat, Eds., vol. 199. Berlin: Springer-Verlag, jun 1994, pp. 144–157.

[7] P. Wuechner, H. de Meer, J. Barner, and G. Bolch, "MOSEL-2 – a compact but versatile model description language and its evaluation environment," in *Proc. MMBnet'05, Hamburg*, B. Wolfinger and K. Heidtmann, Eds. University of Hamburg, sep 2005, pp. 51–59.

[8] G. Di Marzo Serugendo, N. Foukia, S. Hassas, A. Karageorgos, S. K. Mostéfaoui, O. F. Rana, M. Ulieru, P. Valckenaers, and C. Van Aart, "Self-organisation: Paradigms and applications," in *Proc. 11th International Conference on Analysis and Optimization of Systems – Discrete Event Systems*, ser. Lecture Notes in Computer Science, vol. 2977. Springer-Verlag, 2004, pp. 1–19.

[9] H. De Meer and C. Koppen, "Characterization of self-organization," in *Peer-to-Peer Systems and Applications*, ser. Lecture Notes in Computer Science, R. Steinmetz and K. Wehrle, Eds., vol. 3485. Springer-Verlag, 2005, pp. 227–246.

[10] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, jan 1982.

[11] H. De Meer and C. Koppen, "Self-organization in peer-to-peer systems," in *Peer-to-Peer Systems and Applications*, ser. Lecture Notes in Computer Science, R. Steinmetz and K. Wehrle, Eds., vol. 3485. Springer-Verlag, 2005, pp. 247–266.

[12] J. Ledlie, J. Taylor, L. Serban, and M. Seltzer, "Self-organization in peer-to-peer systems," in *Proc. 10th European SIGOPS Workshop*, September 2002.

[13] M. Rosvall and K. Sneppen, "Modeling self-organization of communication and topology in social networks," *Phys. Rev. E*, vol. 74, no. 1, p. 016108, jul 2006.

[14] T. Kohonen, *Self-Organizing Maps*, 3rd ed., ser. Springer Series in Information Sciences. Springer-Verlag, 2001, vol. 30.

[15] T. M. Liggett and S. W. W. Rolles, "An infinite stochastic model of social network formation," *An infinite stochastic model of social network formation*, vol. 113, no. 1, pp. 65–80, sep 2004.

[16] J. B. Schmitt and U. Roedig, "Sensor network calculus Ű a framework for worst case analysis," in *Proc. First IEEE International Conference on Distributed Computing in Sensor Systems,*, ser. Lecture Notes in Computer Science, vol. 3560, IEEE. Springer-Verlag, 2005, pp. 141–154.

[17] T. H. Labella, G. Fuchs, and F. Dressler, "A simulation model for self-organised management of sensor/actuator networks," in *Proc. GI/ITG KuVS Fachgespräch Selbstorganisierende, Adaptive, Kontextsensitive verteilte Systeme (SAKS)*, mar 2006.

[18] J. Starzyk and Z. Zhu, "Software simulation of a self-organizing learning array system," in *Proc. 6th IASTED Int. Conf. Artificial Intelligence & Soft Comp. (ASC 2002)*, H. Leung, Ed. IASTED/ACTA Press, jul 2002.

[19] D. Stamatelakis and W. D. Grover, "Opnet simulation of self-organizing restorable sonet mesh transport networks," in *Proc. OPNETWORKS '98 Conference (CD-ROM)*, apr 1998, p. paper 04.

[20] M. Sethson, P. Krus, and M. Karlsson, "Simulation of self organizing structures using neuro mechanical networks," in *Foundations for Successful Modelling & Simulation : Proc. 17th European Simulation Multiconferece (Esm2003)*, D. Al-Dabass, Ed., jun 2003, pp. 532–537.

[21] J. Cao, "Self-organizing agents for grid load balancing," in *Proc. 5th IEEE/ACM International Workshop on Grid Computing (GRID 2004)*. IEEE/ACM, nov 2004, pp. 388–395.

[22] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, 2nd ed. John Wiley & Sons, nov 2001.