Expert Systems With Applications 44 (2016) 230–244

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Towards modeling and runtime verification of self-organizing systems

Bahareh Abolhasanzadeh*, Saeed Jalili

Electrical and Computer Engineering Faculty, Tarbiat Modares University, Tehran 14115194, Iran

ARTICLE INFO

Keywords:
Self-organization
Formal modeling
Runtime verification
HPobSAM

ABSTRACT

According to the fact that the intrinsic dynamism of self-organizing systems challenges the existing methods of engineering for modeling reliable complex systems, in this paper, we propose a new formal-based method to model self-organizing systems. The capabilities of the proposed method which are used to address several challenges in design, development and analysis of self-organizing systems are: modularity and robustness, decentralized control and scalability, required adaptation types, flexible and adaptive control mechanism, separation of adaptation and business logic, and safe adaptation. To evaluate the proposed method, we use self-organizing traffic management system as a case study and exploit the proposed method for modeling this dynamic system. Moreover, we propose and employ a novel policy-based runtime verification mechanism to ensure that the safety properties are satisfied by the implementation at runtime. We provide our case study prototype using Java and the Ponder2 toolkit and apply our runtime verification method to show its proper reaction capabilities to the property violations. This benefit is the result of using dynamic policies in our method to control the behavior of systems.

1. Introduction

The continuous advances in technology make computing environments dynamic, heterogeneous and more complex. To deal with such dynamics in the environmental conditions, systems should be given more degrees of autonomy, allowing them to continue operating without human intervention. To this aim, in recent years, there has been a significant growth of interest in new approaches based on bio-inspired metaphors to create powerful methodologies for designing systems and solving computational problems. In nature, biological systems are capable to enlarge in scale and behave autonomously without a centralized control; they are only influenced by local environment conditions and interactions with nearby entities (Seeley, 2002). This natural adaptation and loosely-coupled cooperation is the idea behind self-organizing mechanisms. Generally, self-organization is specified as the evolution of a system into an organized state, in the presence of ever changing environments and in the absence of external or of centralized control. Self-organizing systems are generally made of multiple autonomous components which are able to self-organize their activity patterns toward a specified common goal (Macías-Escrivá, Haber, del Toro, & Hernandez, 2013). Such systems are expected to withstand disturbance and do structural adaptation in response to unexpected situations, where structure can be spatial, temporal, or functional.

Motivation: Two main challenges faced by designers of self-organizing systems are (i) designing the autonomous system elements to achieve the intended system-level dynamics and (ii) providing assurances about the correctness of the emergent global behavior.

Based on the degree of autonomy, three approaches are used for designing complex systems. In *centralized* approach, a system is managed in a top-down way using one control element. This approach comprises initiatives like Soft Computing (Zadeh, 1997). Centralized control imposes a single point of failure and needs a complete description of the desired properties and environment. *Hierarchical* approaches consider several control elements distributed over different levels of hierarchy. These control elements interact with each other to realize the goals. Organic Computing (Muller-Schloer, 2004; Schmeck, Müller-Schloer, Çakar, Mnif, & Richter, 2010) is the most notable initiative of such approaches which provides controlled autonomy for a system. However, this approach limits the system's scalability. Current aims in development of self-organizing systems are mainly concentrated on employing *decentralized* approaches. In a decentralized approach, the control of a system is distributed between its components where the system-wide behavior is emerged from local interactions of its components in a bottom-up way.

Different frameworks and models have been investigated by diverse research groups from different communities for designing self-organizing software systems. Among them agent-based modeling is generally considered a good paradigm to tackle complexity problems of self-organizing systems (Bernon, Chevrier, Hilaire, & Marrow, 2006; Gardelli, Viroli, Casadei, & Omicini, 2008). Considering the agent metaphor (Gershenson, 2007) provides a framework for

* Corresponding author. Tel.: +982188946445.
E-mail addresses: bahareh.abolhasanzadeh@gmail.com (B. Abolhasanzadeh), sjalili@modares.ac.ir (S. Jalili).

http://dx.doi.org/10.1016/j.eswa.2015.09.008
0957-4174/

designing self-organizing systems. This framework is built on the idea of achieving maximum global satisfaction by reducing the local friction. In Sloman and Lupu (2010) a robust software architecture for designing autonomic systems is proposed and policy-based computing has been selected as the core self-management technology. Furthermore, biologically inspired principles and mechanisms have recently seen a rapid increase research interest in the context (Villalba & Zambonelli, 2011; Yeom, 2010). Despite the inherent flexibility and capability of these models for supporting properties of self-organization, lack of formal foundation is their great defect.

Formal models are one of the most important aspects of designing complex systems, used to ensure reliability of such systems and to guarantee the functional correctness, safety, and security of them. The intrinsic dynamism of self-organizing systems requires us to use formal methods at early stages of design process to ensure that the system is operating correctly. These approaches provide a systematic way for designing complex systems, and support the validation and verification of system's adaptation before implementation (Adler, Schaefer, Schuele, & Vecchié, 2007).

In spite of the fact that there is a considerable works on formal modeling of dynamic adaptation for complex systems, most of them do not support properties of self-organizing systems. For instance, (Bradbury, Cordy, Dingel, & Wermelinger, 2004; Hongzhen & Guosun, 2010; Wermelinger & Fiadeiro, 2002) only focus on system dynamic reconfiguration and description of dynamic changes at high levels of abstraction, whereas Hadj-Kacem, Kacem, and Drira (2009) and Zhang and Cheng (2006) concentrate on behavioral adaptation of dynamic systems. However, self-organizing systems are enabled to do structural adaptation. Therefore, it is essential for a model to support both behavioral adaptation and dynamic reconfiguration. So far, only few works pay attention to both aspects of adaptation which are necessary to develop self-organizing systems. Among them Canal, Cámara, and Salaün (2012) introduced a framework for dynamic reconfiguration of components with behavioral adaptation considerations; in this work behavioral adaptation is expressed through component reconfiguration. Also, Costa-Soria (2011) presents a framework which allows for the representation of hierarchical systems with adaptive component behaviors and dynamic reconfiguration.

Another aspect of modeling self-organizing systems is the flexibility and scalability of the model. Most of the existing formal models have not focused on these features which are necessary to deal with complexity and evolution of such systems. For instance, graph transformation is used for modeling both dynamic reconfiguration and behavioral adaptation (Becker & Giese, 2008), but it is not flexible to address the requirements of modeling the long-term evolution of systems and complex environments.

One of the most significant aspects of developing self-organizing systems is the challenge of providing assurance that the emergent behaviors adhere to key properties and the system operates correctly during and after reorganizations. Given a model of the system and a desired specification, formal verification aims to prove the correctness of design according to the defined specification. There has been a significant amount of work and different techniques devoted to formal verification of complex systems, such as model checking (Clarke, Grumberg, & Peled, 1999), theorem proving (Cook, 1971), etc. These techniques play an important role in validation and verification of complex systems, but the complexity of self-organizing systems, as well as their subsequent evolution makes verification of such systems a great deal with increasing difficulty.

Contribution: To address the shortcomings of existing methods, we propose a new and flexible method for design and verification of self-organizing systems and discuss different capabilities of this model in supporting key properties of such systems. Our method is based on the Hierarchical PobSAM (HPobSAM) (Khakpour, Jalili, Sirjani, Goltz, & Abolhasanzadeh, 2012) which is a formal model for developing, modeling and verification of self-adaptive evolving systems. The main characteristics of the proposed method are (i) modularity and robustness, (ii) decentralized control and scalability, (iii) structural adaptation, (iv) flexible and adaptive control mechanism, (v) separation of the adaptation logic from the business logic and (vi) safe adaptation. We use the capabilities of HPobSAM and consider a system as a set of autonomous components which use policies to govern and adapt themselves. Behavioral adaptation and dynamic reconfiguration is supported through dynamically changing and modifying the policies.

To clarify our method in designing self-organizing systems, we have designed a specific instance of self-organizing traffic management system as a case study. Such systems have a dynamic nature and exhibit emergent properties. Moreover, they are composed of multiple autonomous components which have continuous interactions among themselves and try to accomplish a common goal in a self-organizing manner. Such properties as well as the increasing importance of Intelligent Transportation Systems (ITSs) make self-organizing traffic management system a suitable case study for our paper.

Furthermore, we propose the employment of a novel runtime verification (Bauer, Leucker, & Schallhart, 2006; Montali, 2010) method to verify self-organizing systems at runtime and show its ability to handle violation of properties dynamically. Runtime verification is a solution to avoid formal verification difficulties (e.g., state explosion problem) and acts as a complement to static verification techniques. This kind of verification is an effective approach to ensure that the desired behavior of a system is met at runtime. In particular, since self-organizing systems often operate in unpredictable environments, it is infeasible to model the system behavior before deployment. Therefore, using runtime verification technique to assure the correctness of emergent behaviors, in self-organizing systems is inevitable.

To illustrate the effectiveness of our runtime verification method and to show how it could react properly even in the time of violation, we have implemented our case study, using Java and the Ponder2 toolkit (Twidle, Dulay, Lupu, & Sloman, 2009), and have applied the runtime verification method on this system.

The rest of this paper is organized as follows: In the following section we give an overview of related work. Section 3 describes HPobSAM, briefly. In Section 4 we present our method for modeling self-organizing systems and discuss its capabilities in modeling such systems. We introduce and model our case study, self-organizing traffic management system, in Section 5. Section 6 introduces our method for run-time verification and in Section 7 we conclude the paper.

2. Related work

The presented work opens different perspectives which are related to the modeling, development and runtime verification of the next generation systems. Here, our objective is to present the diversity of approaches that are presented in literature and their appropriateness for modeling and runtime verification of self-organizing systems.

2.1. Biologically inspired approaches

In this context different frameworks and models have been investigated by diverse research groups to design complex and large-scale software systems. Nature-inspired approaches and frameworks contrived through observation of natural systems and mechanisms have recently seen a rapid increase research interest (Balasubramaniam, Botvich, Donnelly, Foghlú, & Strassner, 2006; Schmickl, Hamann, Wörn, & Crailsheim, 2009; Villalba & Zambonelli, 2011). Some of these mechanisms which have gained great interest in the context of self-organizing systems are: flocking, foraging, stigmergy,