Chapter 3
Towards Multiple-Layer Self-Adaptations of Multi-Agent Organizations Using Reinforcement Learning

Xinjun Mao
National University of Defense Technology, China

Menggao Dong
National University of Defense Technology, China

Haibin Zhu
Nipissing University, Canada

ABSTRACT

This chapter proposes a multi-agent organization model for self-adaptive software to examine the autonomous components and their self-adaptation that can be occurred at either the fine-grain behavior layer of a software agent or the coarse-grain organization layer of the roles that the agent plays. The authors design two-layer self-adaptation mechanisms and combine them with reinforcement learning together to tackle the uncertainty issues of self-adaptation, which enables software agents to make decisions on self-adaptation by learning at run-time to deal with various unanticipated changes. The reinforcement learning algorithms supporting fine-grain and coarse-grain adaptation mechanisms are designed. In order to support the development of self-adaptive software, the software architecture for individual agents, the development process and the software framework are proposed. A sample is developed in detail to illustrate our method and experiments are conducted to evaluate the effectiveness and efficiency of the proposed approach.

INTRODUCTION

Our society highly depends on complex IT systems created by integrating and orchestrating independently managed systems. The complexity of such systems stems from the great number of the system's components, the dynamic relationships between them, and the continuous interactions between the system and its varying environment (Sommerville et al, 2012). Software systems in such a kind require self-adaptation in order to deal with unexpected internal and external changes and thereby to guarantee the flexibility, robustness, reliability of the system (Litoiu et al, 2017; Mahdavi-Hezavehi et al, 2017). Self-adaptive software is normally designed as a closed-loop system that aims to adjust various artifacts or attributes in response to changes in the 'self' and/or within the context (Cheng et al, 2009). Here, 'self' means the whole body of the software, mostly implemented in several layers, while the context encompasses everything in the operating environment (Salehie et al, 2009). Such systems are currently required in many application fields such as the CPS, enterprise, industry, (Camara et al, 2016; Camara et al, 2016)

Several challenges must be addressed when developing complex self-adaptive software, including abstraction, modeling, mechanism design, analysis method, software architecture, language and platform. In the past two decades, the software engineering community devotes to investigating innovative ways of developing, deploying, managing and evolving self-adaptive software systems (Cheng et al, 2009; Salehie et al, 2009). Various software engineering technologies for self-adaptive system have been proposed such as software architecture, control loop, running strategy and model, modelling and programming languages (Cheng et al, 2009; Camara et al, 2016; Weynes, 2017).

However, the evolution of software features and the increasing complexity of current software systems challenges existing software engineering technologies and at the same time pose several open issues to develop complex self-adaptive software.

First, many software systems over the Internet are ultra-large scale systems or coalition of systems-of-systems (Somerville et al, 2012; Weynes, 2017). Typically, these systems are composed of a great number of independent and autonomous components, and the whole systems are managed in a decentralized way. Therefore, traditional methods that assume the system can be controlled in a centralized way seems unsuitable. To investigate the self-adaption in such kind of systems and realize the software needs appropriate abstractions and models.

Second, self-adaptation in a software system can occur at different levels simultaneously to satisfy design objectives, ranging from low-level and fine-grain data and behaviors self-adaptation, to high-level and coarse-grain goal and responsibilities self-adaptation of an individual component, and even the structure-level adaptation

Related Content

An Advance Q Learning (AQL) Approach for Path Planning and Obstacle Avoidance of a Mobile Robot

Arpita Chakrabortyand Jyoti Sekhar Banerjee (2013). *International Journal of Intelligent Mechatronics and Robotics (pp. 53-73).*

www.irma-international.org/article/an-advance-q-learning-aql-approach-for-path-planning-and-obstacle-avoidance-of-a-mobile-robot/87481

Modular Assembly Micro-Robots for Natural Orifice Transluminal Endoscopic Surgery: The Future of Minimal Invasive Surgery

Apollon Zygomalas, Kostas Giokasand Dimitris Koutsouris (2014). *Robotics: Concepts, Methodologies, Tools, and Applications (pp. 1600-1612).*

www.irma-international.org/chapter/modular-assembly-micro-robots-for-natural-orifice-transluminal-endoscopic-surgery/84967

The Semantic Dominance of Emotional Templates in Cognitive Structures

Tom Adi (2015). *International Journal of Synthetic Emotions (pp. 1-13).*

www.irma-international.org/article/the-semantic-dominance-of-emotional-templates-in-cognitive-structures/160800

In Defense of Dominance: PAD Usage in Computational Representations of Affect

Joost Broekens (2012). *International Journal of Synthetic Emotions (pp. 33-42).*

www.irma-international.org/article/defense-dominance-pad-usage-computational/66088

Automatic, Dimensional and Continuous Emotion Recognition

Hatice Gunesand Maja Pantic (2012). *Creating Synthetic Emotions through Technological and Robotic Advancements (pp. 73-105).*

www.irma-international.org/chapter/automatic-dimensional-continuous-emotion-recognition/65824