

Chapter 2.22

Building Complex Adaptive Systems: On Engineering Self-Organizing Multi-Agent Systems

Jan Sudeikat¹

Hamburg University of Applied Sciences, Germany

Wolfgang Renz

Hamburg University of Applied Sciences, Germany

ABSTRACT

Agent oriented software engineering (AOSE) proposes the design of distributed software systems as collections of autonomous and pro-active actors, so-called agents. Since software applications results from agent interplay in multi-agent systems (MASs), this design approach facilitates the construction of software applications that exhibit self-organizing and emergent dynamics. In this chapter, we examine the relation between self-organizing MASs (SO-MASs) and complex adaptive systems (CASs), highlighting the resulting challenges for engineering approaches. We argue that AOSE developers need to be aware of the possible causes of complex system dynamics, which result from underlying feedback loops. In this respect current approaches to develop SO-

MASs are analyzed, leading to a novel classification scheme of typically applied computational techniques. To relieve development efforts and bridge the gap between top-down engineering and bottom-up emerging phenomena, we discuss how multi-level analysis, so-called mesoscopic modeling, can be used to comprehend MAS dynamics and guide agent design, respectively iterative redesign.

INTRODUCTION

AOSE (Weiß, 2002) is a prominent approach to the development of complicated distributed software systems. *Agents*, that is, autonomous and pro-active entities, are proposed as a basic design and development metaphor. Since highly dynamic and

distributed application domains lend themselves to be understood as collections of collaborating actors, the agent metaphor provides appropriate design abstractions (Jennings, 2001). As the actual software applications result from agent interplay, they allow *decentralized* coordination mechanisms that promise the purposeful construction of systems that self-organize, that is, establish and maintain structures without external control, justifying intensive research activities (e.g., Kephart & Chess, 2003; Müller-Schloer, 2004). Awareness is rising that MAS implementations comprise the inherent potential to exhibit complex systems dynamics, for example, *criticality*, *phase transitions* (Parunak, Brueckner, & Savit, 2004) and *emergent phenomena* (Serugendo, Gleizes, & Karageorgos, 2006) have been observed.

The need to handle complex system dynamics in MASs is attracting increasing attention in AOSE research, as the rising phenomena complicate and challenge conventional top-down development efforts. So state Henderson-Sellers and Giorgini (2005):

...To alleviate this concern of an uncontrolled and uncontrollable agent system wreaking havoc, clearly emergent behavior has to be considered and planned for at the system level using top-down analysis and design techniques. This is still an area that is largely unknown in MAS methodologies... (p. 4)

It has been found that these dynamics can be embedded implicitly in top-down designs, leading to unexpected synchronizations and oscillations that impair system performance of MASs, composed of agents that individually perform as intended (Mogul, 2005; Moore & Wright, 2003; Parunak, & VanderBok, 1997). Simulations are required to identify these phenomena and empiric practices dominate development approaches (e.g., De Wolf & Holvoet 2006; Edmonds, 2004).

In this chapter, sources for CAS phenomena in MASs are discussed. Particularly, we give an

overview on current best practices for the development of self-organizing dynamics in MASs, leading to a novel classification of implementation approaches. While these practices have been classified phenomenological, insights in underlying mechanisms are required (Maes, 1994) to guide the utilization of emergent dynamics in software.

Current approaches to the construction of MASs can be distinguished between top-down development *methodologies* and bottom-up, *experimentation-based* prototyping procedures. Engineering procedures address the purpose-, cost- and time-oriented construction of applications and typically rely on iterative, top-down procedures that start from system requirements, which are agreed with stakeholders, and refine these to implementable system designs (cf. *Toward Development Methodologies* and *Summary* sub-sections). When engineering efforts target complex system dynamics, the design of applications that reliably exhibit the nonlinear system behaviors is inherently intricate (Edmonds, 2004), forcing elaborate simulation cycles conducted by experts. In order to mediate between both development paradigms a CAS inspired multi-level, *mesoscopic* modeling approach is presented and demonstrated that can be used to *redesign* MASs, enforcing the intended dynamics. Complex systems and complexity scientists as well as AOSE practitioners find a discussion of the particular challenges that MAS development efforts face established solutions.

This chapter is structured as follows. In the next section software-engineering approaches for agent-based systems are briefly introduced, followed by a discussion of relationships between MASs and CASs. Afterwards, engineering approaches to self-organizing dynamics in MASs are discussed. In the fifth section, these are classified and in the sixth section a multi-level modeling approach is presented, supporting iterative cycles of bottom-up analysis and top-down (re-)design. Finally, conclusions and prospects for future work are given.

19 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/building-complex-adaptive-systems/36724

Related Content

Transformation of Government and Citizen Trust in Government: A Conceptual Model

Mohamed Mahmood (2019). *Strategic Management and Innovative Applications of E-Government* (pp. 107-122).

www.irma-international.org/chapter/transformation-of-government-and-citizen-trust-in-government/208728

Strategic Information Systems for Competitive Advantage: Planning, Sustainability and Implementation

Gareth Griffiths and Ray Hackney (2001). *Strategic Information Technology: Opportunities for Competitive Advantage* (pp. 185-199).

www.irma-international.org/chapter/strategic-information-systems-competitive-advantage/29766

Decision Support Systems and Decision-Making Processes

Udo Richard Averweg (2010). *Strategic Information Systems: Concepts, Methodologies, Tools, and Applications* (pp. 122-130).

www.irma-international.org/chapter/decision-support-systems-decision-making/36684

A Systemic Framework for Accelerating Collaboration-Centered Knowledge Management Strategy

Johanna Bragge and Hannu Kivijärvi (2012). *International Journal of Strategic Information Technology and Applications* (pp. 15-37).

www.irma-international.org/article/systemic-framework-accelerating-collaboration-centered/63004

Factors Comprising Effective Risk Communication, Decision-Making, and Measurement of IT and IA Risk

Ricardo Daza and Kathleen M. Hargiss (2018). *International Journal of Strategic Information Technology and Applications* (pp. 23-40).

www.irma-international.org/article/factors-comprising-effective-risk-communication-decision-making-and-measurement-of-it-and-ia-risk/213233