A Formal Modelling Framework for Developing Multi-agent Systems with Dynamic Structure and Behaviour

Petros Kefalas[1], Ioanna Stamatopoulou[2] and Marian Gheorghe[3]

[1] Department of Computer Science, CITY College, Thessaloniki, Greece
kefalas@city.academic.gr
[2] South-East European Research Centre, Thessaloniki, Greece
istamatopoulou@seerc.info
[3] Department of Computer Science, University of Sheffield, UK
M.Gheorghe@dcs.shef.ac.uk

Abstract. Multi-agent systems exhibit highly dynamic behaviour within dynamic environments. Modelling of individual agents within such systems demands considering both evolving data structures and the control over their internal changing states. In addition, modelling of the overall system implies modelling of the agents' configuration, including their ability to exchange messages as well as the ability to re-structure their formation over time. This paper presents a formal modelling framework based on Communicating X-machines, allowing the specification of multi-agent software systems with a dynamic structure and behaviour. A case study illustrates the proposed modelling approach.

1 Introduction

Multi-agent system complexity is due to substantial differences in attributes between their individuals, high computational power required for the processes within agents, non-trivial type or volume of data manipulated by these processes and considerable amount of communication in order to achieve coordination and collaboration. The use of a computational framework that is capable of modelling both the dynamic aspect (change) and the static aspect (data and knowledge), will facilitate modelling and simulation of such complex systems.

The majority of models created for biological or biology-inspired multi-agent systems are based on an assumed, fixed system structure that is not realistic. Our contribution is to show how most of the modelling requirements are captured through the use of a distributed state-based formal method, namely Communicating X-machines. In particular, we propose an extension of Communicating X-machine Systems, which includes the rules under and operations with which a multi-agent system changes its configuration over time.

Our motivating example is given in Sect. 2 of this paper. In Sect. 3, we briefly discuss the use of formal methods in agent-oriented software engineering and we informally present X-machines. Formal definitions are given in Sect. 4 and a