Formal Modelling and Verification of Reactive Agents for Intelligent Control

George Eleftherakis, Anna Sotiriadou and Petros Kefalas

Computer Science Department
City Liberal Studies
Affiliated Institution of the University of Sheffield
13 Tsimiski Str., 54624 Thessaloniki, Greece
Email: {eleftherakis, sotiriadou, kefalas }@city.academic.gr

Abstract— **Intelligent agent engineering has proved to be a difficult task due to the inherent complexity of agent systems. Finite State Machines are too simple to capture the modelling needs of complex agents that normally require manipulation of non-trivial data structures. X-machine is a formal method that provides such a data structure in form of a memory attached to a set of state. On the other hand, finite state machines models are suitable for verification through model checking, i.e. to prove that certain properties are satisfied by a system model. However, with existing logics, it is obscure how one can describe properties that refer to the memory data structure of an X-machine. This paper describes how a new logic, namely XmCTL, which extends temporal logic with memory quantifiers, facilitates model checking of X-machine agent models. The use of XmCTL is demonstrated through the verification of an agent that controls a steam-boiler system.**

Keywords: **Agents, Temporal Logic, Model Checking**

I. Introduction

An agent is an encapsulated computer system that is situated in some environment and is capable of flexible, autonomous action in that environment in order to meet its design objectives [1]. Agents, as highly dynamic systems, are concerned with three essential factors: (i) a set of appropriate environmental stimuli or inputs, (ii) a set of internal states of the agent, and (iii) a set of rules that relate the two above and determines what the agent state will change to if a particular stimulus arrives while the agent is in a particular state.

One of the challenges that emerge in intelligent agent engineering is to develop agent models and agent implementations that are correct. The criteria for correctness, as stated in [2], are: (i) the initial agent model should match with the requirements, (ii) the agent model should satisfy any necessary properties in order to meet its design objectives, and (iii) the implementation should pass all tests constructed using a complete functional test generation method. All the above criteria are closely related to three stages of agent system development, i.e. modelling, verification and testing.

Although agent-oriented software engineering aims to manage the inherent complexity of software systems [3], there is still no evidence to suggest that any methodology proposed leads towards correct systems. In the last few decades, there has been a strong debate on whether formal methods can achieve this goal. Academics and practitioners adopted extreme positions either for or against formal methods [4]. It is, however, apparent that the truth lies somewhere in between and that there is a need for use of formal methods in software engineering in general [5], while there are several specific cases proving the applicability of formal methods in agent development [6], [7], [8], [9], [10].

In this paper, we use a formal method, namely X-machines, to model and verify the properties of reactive agents. X-machines is a formal method which closely suits the needs of agent development [11], offering an intuitive and practical way of modelling [2] and at the same time a formal testing strategy to test the implementation against the X-machine model [12]. We have analytically described elsewhere [13], [14], that X-machines and its extension, namely communicating X-machines, are particularly suitable for modelling of agent systems. Here, we focus on the verification of such agent models.

The paper is organised as follows. Section 2 provides an introduction to X-machine modelling as well as a fairly complex example of a steam-boiler model. A brief overview of model checking through Temporal Logic is presented in Section 3. Section 4 discusses the main issues of model checking X-machines and the new language devised for this purpose. Finally, the last section concludes with a discussion and suggestions for further work.

II. Modelling Reactive Agents through X-machines

Among formal methods, the Finite State Machines (FSM) manage to capture the essential feature of an agent system, which is the change of its internal state. FSM is a rather straightforward way for modelling reactive agents that receive inputs from the environment and act upon these inputs according to their current state. The FSM, however, lacks the ability to model any non-trivial data structures. Using FSM or their variants to implicitly deal with different values is rather complicated, since the number of states increases in combinatorial fashion to the possible values of the memory structure.