A Framework for Automatically Ensuring The Conformance of Agent Designs

Yoosef Abushark, John Thangarajah, James Harland, Tim Miller

PII:	S0164-1212(17)30104-8
DOI:	10.1016/j.jss.2017.05.098
Reference:	JSS 9970

To appear in: *The Journal of Systems & Software*

Received date:	25 January 2017
Revised date:	4 May 2017
Accepted date:	25 May 2017

Please cite this article as: Yoosef Abushark, John Thangarajah, James Harland, Tim Miller, A Framework for Automatically Ensuring The Conformance of Agent Designs, *The Journal of Systems & Software* (2017), doi: 10.1016/j.jss.2017.05.098

A Framework for Automatically Ensuring The Conformance of Agent Designs[1]

Yoosef Abushark[2], John Thangarajah and James Harland

RMIT University, Melbourne, Australia

{first name}.{last name}@rmit.edu.au

Tim Miller

The University of Melbourne, Melbourne, Australia

tmiller@unimelb.edu.au

Abstract

Multi-agent systems are increasingly being used in complex applications due to features such as autonomy, pro-activity, flexibility, robustness and social ability. These very features also make verifying multi-agent systems a challenging task. In this article, we propose a mechanism, including automated tool support, for early phase defect detection by comparing the plan structures of a *Belief-Desire-Intention* agent design against the requirements models and interaction protocols. The basic intuition of our approach is to extract sets of possible behaviour runs from the agents' behaviour models and to verify whether these runs conform to the specifications of the system-to-be or not. This approach is applicable at design time, not requiring source code, thus enabling detection and removal of some defects at an early phase of the software development lifecycle. We followed an experimental approach for evaluating the proposed verification framework. Our evaluation shows that even simple system's specifications developed by relatively experienced developers are prone to defects, and our approach is successful in uncovering most of these defects. In addition, we conducted a scalability analysis on the approach, and the outcomes show that our approach can scale when designs grow in size.

Keywords: agent-oriented software engineering; multi-agent systems; verification

1. Introduction

Autonomous agents are increasingly being used for developing complex systems in a broad range of domains [1]. Agents in such systems are often autonomous, in order to make the most suitable decisions based on the current environment situations, without human intervention. Many models were proposed to be used as blueprints for developing agent-based systems [2]. A popular and successful model that have been used in many application areas such as automated manufacturing, robotics and flight management is the *Belief-Desire-Intention* (BDI) model [3]. The BDI model has been used as the basis for many agent implementation platforms such as JACK, Jason and JadeX [4].

The development of BDI-based systems involves many design and implementation activities hence numerous *Agent-Oriented Software Engineering* (AOSE) methodologies have been proposed to act as frameworks that organise these activities (see [5] for a recent survey). The development activities within these BDI-based methodologies are performed by software engineers (i.e. humans); hence, these activities are error-prone. These errors may result in undetected faults in the final product and consequently, lead to significant failures in its behaviour.

In software engineering, it has long been established that *early* detection and resolution of software defects saves the development cost, especially for large projects [6, 7]. However, there has been little research into verifying agent-based designs against their point-of-reference artefacts[3] (i.e. during the design phase and before the implementation phase). Existing work on verifying BDI agent systems focuses on formal verification techniques [8], particularly using model checking (e.g. [9]) and theorem proving (e.g. [10]), or runtime testing (e.g. [11]). Even though these techniques are effective in ensuring the correctness of the agent system, they are a *late* check, and they require complete agent programs to be written before any verification is performed.

In this article, we propose a framework, grounded in the Prometheus methodology [12], that enables early detection of defects in agent designs. Prometheus is a well-established agent-oriented methodology that has not only been adopted in academia, but also it has been used in the industry [13, 14]. Even though our framework is grounded in Prometheus, the general concepts and approach are applicable to the other methodologies that support the BDI model of agency, as they are similar [15].

[1] Acknowledges Professor Michael Minikoff for his contributions.

[2] Acknowledges King Abdulaziz University for the scholarship.

[3] Such artefacts document the specifications of the intended system