

Multi-Agent System Development MADE Easy

Zhiqi Shen¹, Han Yu¹, Chunyan Miao¹, Siyao Li¹, Yiqiang Chen²

¹Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly, Nanyang Technological University, Singapore

²Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
 {zqshen, han.yu, ascymiao, lisi0010}@ntu.edu.sg, yqchen@ict.ac.cn

Abstract

Agent-Oriented Software Engineering (AOSE) is an emerging software engineering paradigm that advocates the application of best practices in the development of Multi-Agent Systems (MAS) through the use of agents and organizations of agents. This paper outlines the MADE system, which provides an interactive platform for people who are not well-versed in AOSE to contribute to the rapid prototyping of MASs with ease.

Introduction

Agent Oriented Software Engineering (AOSE) is a software engineering paradigm that advocates the application of best practices in the development of Multi-Agent Systems (MAS) through the use of agents and organizations of agents. Over the years, intelligent agents have evolved from software which responds to stimuli (i.e., reactivity) to programs which take initiative (i.e., autonomy) to fulfill their goals (i.e., proactiveness) and interact with each other (i.e., social ability) (Wooldridge and Jennings 1995).

Currently, agent development tools are mostly based on sophisticated agent modelling methodologies such as the *Belief-Desire-Intention* (BDI) model (Padgham and Winikoff 2004). Thus, agent developers often require specialized knowledge or training. However, as agents often need to be incorporated into real-world applications, the ability to infuse knowledge from domain experts is key to their success (e.g., in (Shieh et al. 2012), knowledge from security personnel is needed). How to enable domain experts who may not be familiar with AOSE to easily contribute to agent development is a significant research challenge.

To address this problem, we propose the Multi-Agent Development Environment (MADE) (Yu, Shen, and Miao 2008). It is based on Goal Net which is a goal-oriented agent mental state modelling methodology (Shen, Miao, and Gay 2006). Users can graphically design and manipulate the mental model of an agent using the Goal Net Designer (GND) integrated development environment (IDE) under MADE to observe the effects of their design decisions in real-time. The agent designs are captured by the Goal Net model. Agent developers only need to provide program

Copyright © 2016, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

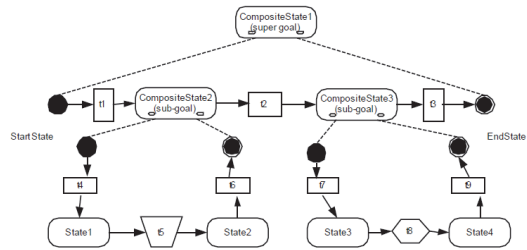


Figure 1: An example Goal Net.

codes governing specific functions the agent should perform when transiting from one goal to the next goal. In this way, MADE reduces the need for users to code agent internals and opens up new possibilities for people with limited AOSE knowledge to develop MASs.

The Goal Net Methodology

Goal Net can be used to support the entire AOSE lifecycle. Its main design artifacts are simplified into three categories based on human intuitions to facilitate easy understanding by non-technical people. They are *goals* (a.k.a. states), *transitions* and *arcs* (as shown in Figure 1). A goal (i.e., circles and round rectangles in Figure 1) denotes the immediate milestone an agent needs to achieve during its reasoning process. Goals can be iteratively decomposed into sub-goal-nets. A transition defines a set of actions that enables an agent to transit from one goal to another. Depending on the types of transitions, they are represented as rectangles, hexagons, or inverted trapezoids (Figure 1). An arc is a design artifact that indicates the direction of logic flow in an agent’s mental state. Readers who are interested in more details regarding the semantics of Goal Net can refer to (Shen, Miao, and Gay 2006).

Designing an Agent

Based on the Goal Net methodology, an IDE called the Goal Net Designer has been developed to enable people with limited knowledge in AOSE to design agents easily (Figure 2). The process of designing an agent using GND begins with a user determining the goals an agent needs to achieve from

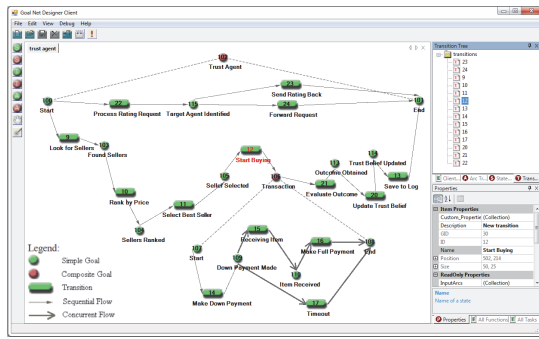


Figure 2: The Goal Net Designer (GND) user interface.

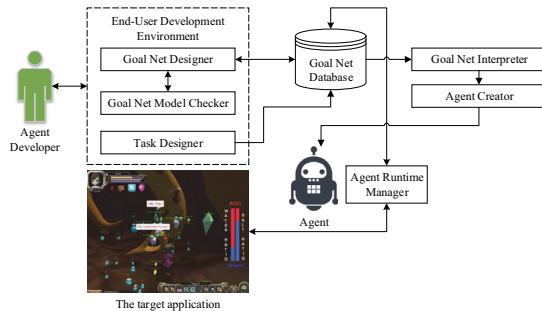


Figure 3: The MADE architecture.

the beginning to the end of a cycle of operation. This step can be easily performed by placing the goals on the IDE interface and labeling them with proper names.

Secondly, variables in each goal can be specified using the IDE to link them with variables in a target system in which the agent will operate. The desirable values for the variables which cause a goal to be regarded as achieved are also required when the user specifies these variables.

Thirdly, the user needs to place transitions between pairs of goals so that the agent knows what tasks need to be carried out in order to transit from one goal to another. Within each transition, the user can specify a list of tasks to be executed in the given sequence. Each task can be implemented as a web service or a function in a dynamic linked library (DLL) so that it can be executed by the agent at runtime.

Lastly, the user connects the goals and transitions of the goal net together using arcs. GND supports both sequential arcs and concurrent arcs (Figure 2). In this way, complex flows of logic represented by a combination of execution sequences can be specified for an agent.

At any stage during the design process, the user can click on the model checking button on the GND IDE to check the correctness of the design. The model checking function automatically checks the design for compliance with the Goal Net semantics, identifies potential deadlocks/livelocks in the logic, and generates a report for the user.

Deploying an Agent

As shown in Figure 3, a user interacts directly with GND. Once an agent is designed, the Goal Net structure data and

the related meta-data are saved into a database on the Cloud. To deploy an agent, the Agent Creator creates an agent based on selected the Goal Net data. The Interpreter of the agent parses the Goal Net data with input from the target application to decide the next goal to pursue and invokes the tasks defined in the transition towards the next goal. The Agent Runtime Manager supports the agent runtime environment, and facilitating the communication between agents and the target applications. Examples of Goal Net agents developed using MADE can be found in (Yu et al. 2010; 2011; Wu et al. 2013; Cai et al. 2014; Yu et al. 2014; Lin et al. 2015). In summary, MADE enables agent developers to design agents graphically and develop the required tasks in the transitions via traditional software engineering.

Acknowledgements

This research is supported, in part, by the National Research Foundation, Prime Minister’s Office, Singapore under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office; and the Lee Kuan Yew Post-Doctoral Fellowship Grant.

References

Cai, Y.; Shen, Z.; Liu, S.; Yu, H.; Han, X.; Ji, J.; McKeown, M. J.; Leung, C.; and Miao, C. 2014. An agent-based game for the predictive diagnosis of parkinson’s disease. In *AAMAS’14*, 1663–1664.

Lin, J.; Yu, H.; Miao, C.; and Shen, Z. 2015. An affective agent for studying composite emotions. In *AAMAS’15*, 1947–1948.

Padgham, L., and Winikoff, M. 2004. *Developing Intelligent Agent Systems: A Practical Guide*. John Wiley and Sons.

Shen, Z.; Miao, C.; and Gay, R. 2006. Goal-oriented methodology for agent-oriented software engineering. *IEICE Transactions on Information and Systems* E89(D):1413–1420.

Shieh, E.; An, B.; Yang, R.; Tambe, M.; Baldwin, C.; DiRenzo, J.; Maule, B.; and Meyer, G. 2012. PROTECT: A deployed game theoretic system to protect the ports of the united states. In *AAMAS’12*, 13–20.

Wooldridge, M., and Jennings, N. R. 1995. Intelligent agents: Theory and practice. *The Knowledge Engineering Review* 10(37).

Wu, Q.; Han, X.; Yu, H.; Shen, Z.; and Miao, C. 2013. The innovative application of learning companions in virtual singapura. In *AAMAS’13*, 1171–1172.

Yu, H.; Cai, Y.; Shen, Z.; Tao, X.; and Miao, C. 2010. Agents as intelligent user interfaces for the net generation. In *IUI’10*, 429–430.

Yu, H.; Shen, Z.; Miao, C.; and Tan, A.-H. 2011. A simple curious agent to help people be curious. In *AAMAS’11*, 1159–1160.

Yu, H.; Miao, C.; An, B.; Shen, Z.; and Leung, C. 2014. Reputation-aware task allocation for human trustees. In *AA-MAS’14*, 357–364.

Yu, H.; Shen, Z.; and Miao, C. 2008. A goal-oriented development tool to automate the incorporation of intelligent agents into interactive digital media applications. *ACM Computers in Entertainment* 6(2):24:1–24:15.