Using Multi-objective Optimization to Generate Timely Responsive BDI Agents

Doctoral Consortium

Márcio Fernando Stabile Junior
Universidade de São Paulo
São Paulo, Brazil
mstabile@ime.usp.br

ABSTRACT

A BDI agent's ability to perform well depends on its reasoning time. If the reasoning is slow, it is possible that the environment has changed and the action selected is no longer optimal by the time the agent has finished to deliberate. This work then builds a BDI architecture using Anytime Algorithms that can control the amount of time used by the agent to reason and act on the environment. I briefly describe the proposed architecture and its implementation in the Jason agent language.

KEYWORDS

BDI agents; Anytime algorithms; Multi-objective optimization

ACM Reference Format:
Márcio Fernando Stabile Junior. 2022. Using Multi-objective Optimization to Generate Timely Responsive BDI Agents: Doctoral Consortium. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

1 INTRODUCTION

The BDI model proposed by Bratman [2] and formalized by Rao and Georgeff [6] mainly consists of reviewing the agent's beliefs based on perceptions, listing and filtering desires to generate intentions, and creating and executing plans to achieve those intentions. However, Bratman et al. [3] states that one of the problems with agents is that reasoning is not immediate, as studied by Simon [8]. Hence, the agent risks that the environment changes and the selected action is no longer optimal when the agent finishes deliberating. Considering this problem, to plan actions in scenarios where the time available to generate the plan is variable, and the process of decision making is complex, Dean and Boddy [4] defined a class of algorithms called *anytime algorithms*. Based on the work of Dean and Boddy [4], Zilberstein [12] proposed ways to analyze and compose algorithms in order to create complex systems that allow the balancing between processing time and quality of results. Since one of the issues of the BDI architecture is the need for fast practical reasoning, and the main characteristic of Anytime Algorithms is the possibility of controlling the execution time, it seems logical to use anytime algorithms to design more efficient BDI agents.

This work then aims to answer the question: How can we build a BDI architecture using Anytime Algorithms that can control the amount of time used by the agent to reason and act on the environment while assuring a minimum quality in the actions?

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

2 ANYTIME BDI

In order to answer this question, in this section I present a brief description of the formal model called Anytime BDI. First, it is necessary to define three terms: Internal actions, external actions and plans. Based on the definitions made by Schut et al. [7], I formally define these elements below:

(1) An external action (α) is an action that is performed by the agent on the environment. Examples of external actions include agent movement, activating a button and carrying an item. (2) An internal action (β) is an action that is performed by the agent that does not directly change the environment. Instead, it affects the internal state of the agent. Examples of internal actions include creating a new belief or acquiring a new goal. (3) A plan (π) is a pre-defined sequence of internal (α) and external (β) actions.

Having made these definitions, I present a description of the model of the Anytime BDI Agent:

An Anytime BDI Agent is an intelligent agent architecture composed of two layers. One is the Agent Data Layer (ADL), which contains all the agent's data structures. The other is the Agent Control Layer (ACL), composed of the mechanisms that control the agent execution. The Anytime Data Layer is comprised of all the structures that store the necessary data for the execution of the agent. We define $ADL = \langle \mathcal{P}, \mathcal{B}, \Pi, \mathcal{D}, \mathcal{I}, \alpha_{default}, t_\Delta, \mathcal{DA}, \mathcal{PP} \rangle$ where:(1) The set of perceptions $\mathcal{P}$; (2) The set of beliefs $\mathcal{B}$; (3) The set of desires $\mathcal{D}$; (4) The set of plans Π; (5) The set of intentions $\mathcal{I}$; (6) The default action $\alpha_{default}$ executed if no better action was found; (7) The response time t_Δ; (8) The queue of delayed actions $\mathcal{DA}$; (9) The set of performance profiles $\mathcal{PP}$;

The Anytime Control Layer is comprised of the processes responsible for running the Anytime BDI Agent and ensuring its execution within the specified time. We define $ACL = \langle \mathcal{BM}, \mathcal{IG}, \mathcal{IE}, \mathcal{M}, \mathcal{F} \rangle$ where: (1) $\mathcal{BM}$: Belief Manager is the anytime component responsible for producing a new set of beliefs from perceptions; (2) $\mathcal{IG}$: Intention Generator is the anytime component responsible for based on current beliefs, desires, and intentions, produce a new set of intentions and plans; (3) $\mathcal{IE}$: Intention Executor is the anytime component responsible for based on current beliefs, intentions, utility function, and delayed actions, choose an external action to be executed on the environment; (4) $\mathcal{M}$: Monitor is the component responsible for based on the performance profiles and t_Δ, calculate time allocation and control the execution; (5) $\mathcal{F}$ is a function responsible for calculating the utility of an external action α.

The goal of this model is to make the agent perform an external action α in the environment at each time interval t_Δ. For this, three modules were defined ($\mathcal{BM}$, $\mathcal{IG}$, and $\mathcal{IE}$), which are anytime algorithms responsible for analyzing perceptions, beliefs, desires,