

Subgoal Semantics in Agent Programming

M. Birna van Riemsdijk, Mehdi Dastani, and John-Jules Ch. Meyer

ICS, Utrecht University, The Netherlands
{birna, mehdi, jj}@cs.uu.nl

Abstract. This paper investigates the notion of subgoals as used in plans in cognitive agent programming languages. These subgoals form an abstract representation of more concrete courses of action or plans. Subgoals can have a procedural interpretation (directly linked to a concrete plan) or a declarative one (the state to be reached as represented by the subgoal is taken into account). We propose a formal semantics for subgoals that interprets these declaratively, and study the relation between this semantics and the procedural subgoal semantics of the cognitive agent programming language 3APL. We prove that subgoals of 3APL can be programmed to behave declaratively, although the semantics is defined procedurally.

1 Introduction

This paper presents an *observation* about the cognitive agent programming language 3APL [8]. The observation is related to the notion of a *goal*. This is an important concept in cognitive agent programming languages. Goals are introduced to specify an agent's proactive behavior. Many languages and platforms have been proposed to implement (represent and process) an agent's goals [11,8,15,4,16,3,14,2]. The way in which goals are dealt with varies from language to language. In some programming languages goals are interpreted in a *procedural* way as processes that need to be executed. In others goals are interpreted in a *declarative* way as states to be reached. Yet other languages combine both aspects. Procedural goals are also often called *plans*, which is a terminology we will also use in this paper.

While the procedural interpretation might arguably be considered more standard, the declarative interpretation of goals also has several advantages. Most importantly in this context, is the fact that declarative goals provide for the possibility to decouple plan execution (i.e., the execution of a procedural goal) and goal achievement (i.e., the achievement of a declarative goal) [16]. If a plan fails, the goal that was to be achieved by the plan remains a goal of the agent. The agent can then for example select a different plan or wait for the circumstances to change for the better.¹

A common usage of goals, and the one we are concerned with in this paper, is that of *subgoals* as occurring in the plans of the agent.² These plans are often

¹ See e.g. [13] for a more elaborate discussion on the advantages of declarative goals.

² A usage of the term subgoal that we do not consider in this paper is usage in the logical sense, where for example p is considered to be a subgoal of the goal $p \wedge q$ [13].

built from basic actions which can be executed directly, and subgoals which can be viewed as representing a course of action in a more abstract way. An agent can for example have the plan to go to the bus stop, to take the bus into town,³ and then to achieve the goal of buying a birthday cake. This goal of buying a birthday cake will have to be fulfilled by selecting a more concrete plan of for example which shop to go to, etc.

Just as goals in general, subgoals of plans can also be categorized as either procedural or declarative. In the procedural interpretation, subgoals are linked directly to plans. Their only role is the abstract representation of a more concrete plan. In the declarative interpretation, the fact of whether the state that is represented by the subgoal is achieved (for example through the execution of a corresponding concrete plan), is somehow taken into account. In the birthday cake example, this means that it is important whether the execution of the concrete plan of which shop to go to etc., has resulted in a state in which the birthday cake is actually bought. If it turns out that the goal of buying the cake is not reached after having gone to the specific shop, the agent could select another plan to try a different shop. A declarative interpretation of subgoals could yield more flexible agent behavior, because of the decoupling between plan execution and goal achievement.

We thus argue that it is important to be able to express a declarative notion of subgoals in a cognitive agent programming language. This paper aims to investigate whether these declarative subgoals can be expressed in the language 3APL. In order to do this, we first make precise what we mean exactly by declarative subgoals, by defining a simple formal semantics for subgoals that interprets these in a declarative way (sections 2 and 3). We then compare this semantics with the semantics of 3APL (section 4). We show that 3APL has a notion of subgoal, but it is a procedural kind of subgoal. It turns out, however, that although subgoals of 3APL are defined to have a procedural semantics, a 3APL agent can nevertheless be *programmed* to have these subgoals behave *declaratively*. This observation (and a formal proof that it is correct) is the main contribution of this paper.

The 3APL language family [8,15,4] is an example of a set of languages in which subgoals are interpreted procedurally. Languages and platforms from the AgentSpeak family [9,5,11,1,6] also have a procedural view on subgoals, although the mechanism differs from that of 3APL. We conjecture that a similar result for an implementation of declarative subgoals can be obtained for AgentSpeak, although this is left for future research. An example of a declarative view on subgoals is the high-level language of Winikoff et al. [16]. The declarative semantics we propose in section 3 is comparable with that of [16], although [16] has a much more elaborate plan language. An elaborate plan language is however not needed for the purpose of comparison with the procedural subgoals of 3APL. Establishing a formal relation with the work of Winikoff et al. is left for

³ Assuming that both going to the bus stop and taking the bus into town are actions that can be executed directly.