In Proceedings of the Eighth Australian Joint Artificial Intelligence Conference, X. Yao (ed.), 1995

Goal Generation and Adoption in Hierarchical Agent Models

Michael Luck
Dept. of Computer Science
University of Warwick
Coventry, CV4 7AL
United Kingdom
EMAIL mikeluck@dcs.warwick.ac.uk

Mark d'Inverno
School of Computer Science
University of Westminster
New Cavendish Street
London, United Kingdom
EMAIL dinverm@westminster.ac.uk

Abstract

If agents are able to exploit the resources available in a multi-agent domain they must make use of other agents to help them in their tasks. In order to do this it is important that we first of all have an understanding of *agency*, and then of how goals are generated and subsequently adopted by other agents. In this paper we build upon a three-tiered hierarchy which has been constructed to define objects, agents and autonomous agents, where all autonomous agents are agents, and all agents are objects. In this hierarchy, agents are viewed as objects with *goals*, and autonomous agents as agents with *motivations*. Any object may be viewed as an agent if is currently being engaged to some use, and any agent as autonomous if it has the ability to generate its own goals. This view of agency and autonomy is thus based on the generation and transfer of goals between various entities in the world. Specifically, an entity is an agent if it can be viewed as satisfying a goal. This goal must first be created and then, if necessary and appropriate, transferred to another entity. It is this adoption of goals that makes an entity change from an object to an agent, and it is the *self-generation* of goals that defines the autonomy of an agent. We consider the three classes of goal adoption by objects, agents and autonomous agents. The first of these is merely a question of instantiation, the second requires an understanding of the relationship of the agent to the other entities that are engaging it, and the third amounts to a question of negotiation or persuasion. In this paper we specify and describe goal generation and adoption in the context of *hierarchical agent models* using the Z specification language.

Keywords: agency, autonomy, distributed AI, agent models, Z.

1 Introduction

There are many definitions of agents[5, 6]. A recent paper by Wooldridge and Jennings [9] quotes the definition of an agent as "one who, or that which, exerts power or produces an effect."[1] However, they omitted the second sense of agent which is given as "one who acts for another . . ." This is important, for it is not the acting alone that defines agency, but the acting for *someone or something* that is defining. A cup, for example, is an object. We can regard it as an agent, but it serves no useful purpose to do so without considering the circumstances. A cup is an agent *if* it is containing a liquid and it is doing so to some end. In other words, if I fill a cup with coffee, then the cup is my agent — it serves my purpose. It would *not* be an agent if it was just sitting on a table without serving any useful purpose. In this case it would be an object. Note that we do not require an entity to be intelligent for it to be an agent.

[1] *Concise Oxford Dictionary of Current English (7th edition)*, Oxford University Press.

This view of agency is based on the generation and transfer of goals between entities. Specifically, an entity is an agent if it can be viewed as satisfying a goal. This goal must first be created and then, if necessary and appropriate, transferred to another entity. It is this adoption of goals that makes an entity change from an object to an agent, and it is the *self-generation* of goals that is responsible for the autonomy of an agent. In this paper we specify and describe goal generation and adoption in the context of *hierarchical agent models* using the Z specification language. We begin by briefly outlining Luck and d'Inverno's framework for agency and autonomy based on a three-tiered hierarchy of objects, agents and autonomous agents[2]. We describe each of these entities, and show why and how the distinctions between them are both important and useful. Then we describe the role of motivations in goal generation, and finally consider goal adoption and discuss how it helps us (and other agents) to understand interactions in a multi-agent environment.

2 The Agent Hierarchy Framework

The Agent Hierarchy Framework consists of *objects*, *agents* and *autonomous agents*. The basic idea underlying this hierarchy is that all known entities are objects. Of this set of objects, some are agents, and of these agents some are autonomous agents.

Before we can move to a definition of any of these entities, we must first define two primitives the first of these being an *attribute*. Attributes are simply features of the world, and are the only characteristics which are manifest. They need not be perceived by any particular entity, but must be potentially perceivable in an omniscient sense. (The notion of a feature here allows anything to be included.) The second primitive is an *action* which is strongly related to the notion of *agency*.

Definition: An *attribute* is a perceivable feature.

Definition: An *action* is a discrete event which changes the state of the environment.

In Z, before constructing a specification, we must first define types. Here we define the set of all attributes and actions:

$$[Attribute, Action]$$

An object is then defined in terms of its abilities and its attributes with no further defining characteristics. This provides us with the basic building block to develop our notion of agency.

Definition: An *object* comprises a set of actions and a set of attributes.

A state schema can be constructed that defines an object. Z schemas have two parts: the upper, declarative, part which declares variables and their types, and the lower, predicate, part which relates and constrains the variables.

$$\begin{array}{|l} Object \\ \hline attributes : \mathbb{P}\ Attribute;\ capableof : \mathbb{P}\ Action \\ \hline \end{array}$$

An object is an agent if it serves a useful purpose either to a different agent, or to itself, in which case the agent is *autonomous*. Specifically, an agent is something that 'adopts' or satisfies a goal or set of goals (often of another). Thus if I want to store coffee in a cup, then the cup is my agent for storing coffee. It is satisfying, or has been *ascribed*, or has *adopted* my goal, to have the coffee stored. An agent is thus defined in relation to its goals. We take a traditional view of goals as describable environmental states.

Definition: A *goal* is a state of affairs to be achieved in the environment.

$$Goal == \mathbb{P}\, Attribute$$

Definition: An *agent* is an instantiation of an object with an associated set of goals.

$$\begin{array}{|l} Agent \\ \hline Object \\ goals : \mathbb{P}\, Goal \\ \hline \end{array}$$

Thus an agent has, or is *ascribed*, a set of goals which it retains over any instantiation (or lifetime). One object may give rise to different instantiations of agents. An agent is instantiated from an object in response to another agent. Thus agency is *transient*, and an object which becomes an agent at some time may subsequently revert to being an object.

Returning to the cup example, we have an agent with the same attributes and actions as the cup object, but now it can be ascribed the goal — my goal — of *storing my coffee*. Not everyone will know that it is an agent in this way, however. If, for example, I am in a cafe and there is a half-full cup of lemon-tea on my table, there are several views that can be taken. It can be regarded by the waiter as an agent for me, storing my tea, or it can be regarded as an object serving no purpose if the waiter thinks it is not mine. The view of the cup as an object or agent is relevant to whether the waiter will remove the cup or leave it at the table. Note that we are not suggesting that the cup actually possesses a goal, just that there is a goal that it is satisfying.

Consider a robot. If the robot has no goal then it cannot use its actuators in any sensible way but only, perhaps, in a random way, and must be considered an object. Alternatively, if the robot has some goal which allows it to employ its actuators in some directed way, such as picking up a cup, or riveting a panel onto a hull, then it is an agent. The goal need not be explicitly represented, but can instead be implicit in the hardware or software design of the robot. Note that the coffee-cup is passive and has goals *imposed* upon and *ascribed* to it, while the robot is capable of actively manipulating the environment by performing actions designed to satisfy its goals.

This definition of agency relies upon the existence of other agents which provide goals that are adopted in order to instantiate an agent. These agents are *autonomous* agents since they are not dependent on the goals of others. Autonomous agents possess goals which are *generated* from within, rather than *adopted* from, other agents. These goals are generated from *motivations* which are higher-level non-derivative components characterising the nature of the agent, but which are related to goals. Motivations are, however, qualitatively different from goals in that they are not describable states of affairs in the environment. For example, consider the motivation *greed*. This does not specify a state of affairs to be achieved, nor is it describable in terms of the environment, but it may give rise to the generation of a goal to rob a bank. The distinction between the motivation of greed and the goal of robbing a bank is clear, with the former providing a reason to do the latter, and the latter specifying what must be done.

Definition: A *motivation* is any desire or preference that can lead to the generation and adoption of goals and which affects the outcome of the reasoning or behavioural task intended to satisfy those goals. (This draws on the definition used by Kunda [1].)

$$[Motivation]$$

Definition: An *autonomous agent* is an instantiation of an agent together with an associated set of motivations.

$AutonomousAgent$

$Agent$

$motivations : \mathbb{P}\ Motivation$

An autonomous agent with motivations, therefore, has some means of evaluating behaviour in terms of the environment and these motivations, so that its behaviour is determined by both external and internal factors.

In illustration of these ideas, note that the cup cannot be considered autonomous because it cannot generate its own goals. The robot, however, is potentially autonomous in the sense that it may have a mechanism for internal goal generation. Suppose the robot has motivations of achievement, hunger and self-preservation, where achievement is defined in terms of fixing tyres onto a car on a production line, hunger in terms of maintaining power levels, and self-preservation in terms of avoiding system breakdowns. Normally, the robot will generate goals to attach tyres to cars through a series of subgoals. With low power levels, however, it may replace this with a newly-generated goal of its batteries. A third possibility is that in satisfying its achievement motivation, it works for too long and is in danger of overheating. In this case, the robot can generate a goal of pausing for a period to avoid any damage to its components. Such a robot is autonomous because its goals are not imposed, but are generated in response to its environment.

Thus, we have constructed a formal specification which identifies and characterises those entities that are called agents and autonomous agents. Most usefully, perhaps, the specification is constructed in such a way as to allow further levels of specification to be added to describe particular agent designs and architectures.

3 Goal Generation

The three-tiered framework described above involves the generation of *goals* from *motivations* in an autonomous agent, and the adoption of goals by, and in order to create, other agents. In this section, we consider issues in goal generation that must occur before goal adoption can take place. Specifically, we describe how an autonomous agent, *defined* in terms of its high-level and somewhat abstract *motivations*, can construct goals or concrete states of affairs to be achieved in the environment. We extend the framework in this way and add more detail by introducing new schemas that specify the relevant aspects.

Our model requires a repository of known *goals* which capture knowledge of limited and well-defined aspects of the world. These goals describe particular *states* or *sub-states* of the world with each autonomous agent having its own such repository. An agent will try to find a way to mitigate motivations, either by selecting an action to achieve an existing goal, or by retrieving a goal from a repository of known goals. The last of these is considered below.

In order to retrieve goals to mitigate motivations, an autonomous agent must have some way of assessing the effects of competing or alternative goals. Clearly, the goals which make the greatest positive contribution to the motivations of the agent should be selected. The $GenerateGoal$ schema below describes how an autonomous agent monitors its motivations for goal generation. First, the agent changes given by $\triangle AutonomousAgent$. The variable representing the repository of available known goals, $goalbase$ is declared. Then, the motivational effect on an autonomous agent of satisfying a set of new goals is given. The $motiveffect$ function returns a numeric value representing the motivational effect of satisfying a set of goals with a particular configuration of motivations and a set of existing goals. The predicate part specifies that all goals previously and currently being pursued must be

known goals that already exist in the goalbase. The remaining part of the schema states that there is a set of goals in the goalbase that has a greater motivational effect than any other set of goals, and the current goals of the agent are updated to include the new goals.

GenerateGoal

$$\triangle AutonomousAgent$$
$$goalbase : \mathbb{P}\ Goal$$
$$motiveffect : \mathbb{P}\ Motivation \longrightarrow \mathbb{P}\ Goal \longrightarrow \mathbb{P}\ Goal \longrightarrow \mathbb{Z}$$

$$goals \subseteq goalbase \land goals' \subseteq goalbase$$
$$\exists\, gs : \mathbb{P}\ Goal \mid gs \subseteq goalbase \bullet (\forall\, os : \mathbb{P}\ Goal \mid os \in (\mathbb{P}\ goalbase) \bullet$$
$$(motiveffect\ motivations\ goals\ gs \geq motiveffect\ motivations\ goals\ os)$$
$$\land\ goals' = goals \cup gs)$$

4 Goal Adoption

Since we are interested in multi-agent worlds, we must consider the world as a whole rather than just individual agents. In this world, all autonomous agents are agents and all agents are objects. We also identify further sub-categories of entity. Before proceeding, therefore, we distinguish those objects which are not agents, and those agents which are not autonomous and refer to them as *neutral-objects* and *server-agents* respectively.

An agent is then either a server-agent or an autonomous agent, and an object is either a neutral-object or an agent.

NeutralObject

$$Object$$

$$goals = \{\}$$

ServerAgent

$$Agent$$

$$motivations = \{\}$$

We can then describe the world as a collection of neutral objects, server agents and autonomous agents.

World

$$autoagents : \mathbb{P}\ AutonomousAgent$$
$$neutralobjects : \mathbb{P}\ NeutralObject$$
$$serveragents : \mathbb{P}\ ServerAgent$$

In the description given in the previous section, goals may be generated only by autonomous agents. Both non-autonomous (server) nd autonomous agents, however, can adopt goals. With autonomous agents, goal adoption amounts to a problem of *negotiation* or *persuasion*, requiring an analysis of the *target* autonomous agent. With non-autonomous agents, goal adoption requires an analysis of

both the agent intended to adopt the goal, and any other agent *engaging* that agent. With objects, no analysis is required, since agents are *created* from objects with the relevant associated goals.

For explication we distinguish three kinds of agent. A *target* agent or object is one that is intended to adopt goals. An *engaging* agent is one whose goals are currently (already) adopted by the target agent. A *viewing* agent is an agent that seeks to engage a target agent or object by having it adopt goals. It is a viewing agent because the way in which goal adoption is attempted is determined by its view of the situation. We consider the three cases of goal adoption by continuing with examples involving cups. These examples could also have used robots, but cups demonstrate that the model applies equally to entities with no intelligence, or with unknown intelligence.

In the simplest case, goal adoption by non-autonomous agents occurs by instantiating an agent from a neutral object with the goals to be adopted. In this case, no *agent* exists before the goals are adopted, but the act of goal transfer causes an agent to be created from a neutral object using those particular goals. Thus, for example, the cup on my desk, which is just an object, becomes an agent when I use it for storing my coffee, when it *adopts* or *is ascribed* my goal of storing coffee. It is only possible to create the agent from the object because the cup is not being used by anyone else — it is not *engaged* by another agent. An entity can only be a neutral object if it is not *engaged*. Below we specify a function that creates a server-agent by ascribing a set of goals to some existing neutral-object.

$$ObjectAdoptGoals : (NeutralObject \times \mathbb{P}\ Goal) \rightarrowtail ServerAgent$$

$$\forall\, gs : \mathbb{P}\ Goal;\ old : NeutralObject;\ new : ServerAgent \bullet$$
$$ObjectAdoptGoals(old, gs) = new \Leftrightarrow new.goals = old.goals \cup gs$$
$$\wedge\ new.capableof = old.capableof \wedge new.attributes = old.attributes$$

We now specify how a non-autonomous disengaged object, or neutral-object is instantiated as a (server) agent. In Z, a variable with a '?' indicates an input. Thus, an object and a set of goals are input, the entities in the world change, indicated by $\triangle World$, and the sets of objects and agents are updated accordingly.

ObjectAdoptGoalsWorld

$o? : NeutralObject;\ gs? : \mathbb{P}\ Goal$
$\triangle World$

$neutralobjects' = neutralobjects \setminus \{o?\}$
$serveragents' = serveragents \cup \{ObjectAdoptGoals\,(o?, gs?)\}$
$autoagents' = autoagents$

If the cup was *engaged* by another (possibly non-autonomous) agent, then it is itself an agent, and the protocol for goal adoption changes. In this case, there are alternative ways for me to *engage* the cup. The first of these involves me trying to persuade the engaging agent to release the cup so that I may then subsequently engage it for my purposes. This relates to the issues of goal adoption for autonomous agents which are considered later. The second involves supplying the agent with more goals, so that the agent is shared between different engaging agents. The third possibility involves *displacing* the engaging agent so that I become the engaging agent and ascribe to the cup my own goals. For example, the cup may currently be used as a paper-weight for my office-mate, and is therefore her agent with her goal of securing loose papers. I can displace the goal ascribed to the cup by removing the cup and pouring my coffee into it. Now the cup is ascribed my goal of storing coffee, and it has switched from one agent to another. In fact, this is equivalent to the agent reverting to an object and then being re-instantiated as a new agent. This method may not be an appropriate

strategy, however, because in destroying the agency of the cup as a paper-weight, I risk a conflict with the existing engaging agent, my office-mate. It would be better for me to negotiate first, to obtain permission to destroy the original agency. Our notion of agency thus contributes to a better understanding of the world, regardless of whether we are concerned with cups or robots, since the only important difference between them is their functionality through their agency. The mathematical formalism of this operation is similar to that of the previous schema.

With autonomous agents, goals must explicitly be adopted, as opposed to an implicit ascription of goals for non-autonomous agents. This may be more difficult than the previous case, since it requires some form of negotiation. Autonomous agents are motivated agents and will only participate in an activity and assist another agent if it is to their motivational advantage to do so. In the schema below we merely have that the set of new goals which the agent adopts are the best that it can find in its goalbase at that time. It makes use of the function $AutonomousAgentAdoptGoals$ which adds $gs?$ to the existing goals of the agent.

$$
\begin{array}{|l}
\hline
AutonomousAgentAdoptGoalsWorld \\
aa? : AutonomousAgent;\ gs? : \mathbb{P}\ Goal \\
\triangle World \\
\hline
autoagents' = autoagents \setminus \{aa?\} \cup \{AutonomousAgentAdoptGoals\,(aa?, gs?)\} \\
serveragents' = serveragents \\
neutralobjects' = neutralobjects \\
\neg\ (\exists\, hs : \mathbb{P}\ Goal \mid hs \subseteq goalbase \land hs \neq gs? \bullet \\
\qquad motiveffect\ motivations\ goals\ hs > motiveffect\ motivations\ goals\ gs?) \\
\hline
\end{array}
$$

Though we have not specified the way in which motivations change in response to the environment, nor how agents negotiate to achieve desired goal adoption in other agents, we have shown how these notions of agency and autonomy can be used to provide a structure that allows a better understanding of the entities in the world so that negotiation can be effective and efficient.

5 Conclusions

The notion of motivation is not new. Simon, for example, takes motivation to be "that which controls attention at any given time," [7]. Sloman [8] has elaborated on Simon's work, showing how motivations are relevant to emotions and the development of a computational theory of mind. Others have used motivation and related notions in developing computational architectures for autonomous agents such as the *motives* of Norman and Long [4], and the *concerns* of Moffat and Frijda [3]. What is new about the current work is the role of motivation in defining autonomy.

The three-tiered hierarchy distinguishes clearly between objects, agents and autonomous agents in terms of goals and motivations. Such an analysis of the entities in the world not only provides appropriate structures so that different levels of functionality may be established, but also information as to how multiple entities or agents can cooperate to solve problems which could not be solved alone. By basing the distinctions on function and purpose, we do not arbitrarily differentiate between cups and robots, for example, especially when it is not useful to do so. Instead, our motivation and goal based analysis allows us to concentrate precisely on important aspects of multi-agent interaction and problem-solving. In that context, we have considered the roles of goal generation and adoption. We have specified how and why goals must be generated in some autonomous agents in response to motivations, grounding chains of goal adoption, and further, how goals are adopted by objects,

agents and autonomous agents in the hierarchical agent model. This specification thus outlines what is necessary for effective multi-agent systems.

References

[1] Z. Kunda. The case for motivated reasoning. *Psychological Bulletin*, 108(3):480–498, 1990.

[2] M. Luck and M. d'Inverno. A formal framework for agency and autonomy. In *Proceedings of the First International Conference on Multi-Agent Systems*, pages 254–260. AAAI Press / MIT Press, 1995.

[3] D. Moffat and N. H. Frijda. An agent architecture: Will. In *Proceedings of the 1994 Workshop on Agent Theories, Architectures, and Languages*, 1994.

[4] T. J. Norman and D. Long. A proposal for goal creation in motivated agents. In *Proceedings of the 1994 Workshop on Agent Theories, Architectures, and Languages*, 1994.

[5] D. Riecken. An architecture of integrated agents. *Communications of the ACM*, 37(7):107–116, 1994.

[6] T. Selker. A teaching agent that learns. *Communications of the ACM*, 37(7):92–99, 1994.

[7] H. A. Simon. Motivational and emotional controls of cognition. In *Models of Thought*, pages 29–38. Yale University Press, 1979.

[8] A. Sloman. Motives, mechanisms, and emotions. *Cognition and Emotion*, 1(3):217–233, 1987.

[9] M. J. Wooldridge and N. R. Jennings. Agent theories, architectures, and languages: A survey. In *Proceedings of the 1994 Workshop on Agent Theories, Architectures, and Languages*, 1994.