A Motivation-based Approach for Autonomous Generation and Ranking of Goals in Artificial Agents

Luís Macedo[1,2]
[1]Department of Informatics and Systems Engineering, Engineering Institute, Coimbra Polytechnic Institute
R. Pedro Nunes, Quinta da Nora, 3030-199 Coimbra - Portugal
lmacedo@isec.pt

Amílcar Cardoso[2]
[2]Centre for Informatics and Systems of the University of Coimbra
Pinhal de Marrocos, 3030 Coimbra - Portugal
{macedo,amilcar}@dei.uc.pt

Abstract

In this paper we describe an architecture of an artificial agent that is able to autonomously generate and rank its own goals or intentions based on its motivations. We present an experiment conducted in a simulated environment with such an agent.

1 Introduction

Considered by many authors as the principal motivational system, emotion is one of the sub-systems that compose personality (Izard, 1991), a characteristic that agents may exhibit (Etzioni & Weld, 1995). Another important sub-system is the drive system (also an important kind of the motivational system). Psychological and neuroscience research over the past decades suggests that emotions play a critical role in decision-making, action and performance, by influencing a variety of cognitive processes (e.g., attention, perception, planning, etc.). Actually, on the one hand, recent research in neuroscience (Damásio, 1994) supports the importance of emotions on reasoning and decision-making. On the other hand, there are a few theories in psychology relating motivations (including drives and emotions) to action (Izard, 1991). For instance, in the specific case of emotions, within the context of the belief-desire theories of action (the dominant class of theories in today's motivation psychology) there have been proposals (Reisenzein, 1996) such as that emotions are action goals, that emotions are or include action tendencies, that emotions are or include goal-desires, and that emotions are mental states that generate goal-desires.

Another important characteristic that agents should also exhibit is autonomy (Etzioni & Weld, 1995). In order to be autonomous, agents should be able to generate their own goals and state preferences between them.

In this paper we describe an artificial agent that is able to autonomously generate and rank its own goals or intentions based on its motivations.

The next section presents an overview of the agent's architecture, giving special attention to the deliberative reasoning/decision-making module in which the generation and ranking of goals are included. Finally, a qualitative experiment is described, discussed and some conclusions are achieved.

2 Agent's Architecture

The architecture that we adopted for an agent (Figure 1) is based on the belief, desire, and intention (BDI) approach (Rao & Georgeff, 1995). Besides, the agent is of motivational kind, exhibiting a module of emotions, drives and other motivations. These play a central role in reasoning and decision-making since they may be thought as action goals (Reisenzein, 1996). The next subsections describe in more detail the main modules of the architecture. The information of the environment is provided to these modules by the sensors, and the effectors execute the actions selected.

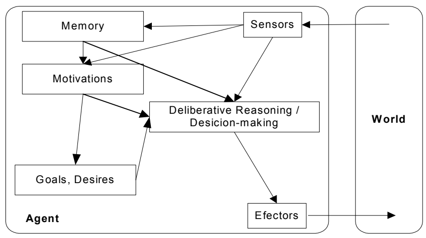

Figure 1. Architecture of an agent.

2.1 Memory

The memory of an agent stores information about the world. This information includes the configuration of the surrounding world such as the position of the entities (objects and other animated agents) that inhabit it, the description of these entities themselves, descriptions of the sequences of actions (plans) executed by those entities and resulting from their interaction, and, in generally, beliefs about the world. This information is stored in several memory components. Thus, there is a metric (grid-based) map (Thrun, 2002) to spatially model the surrounding physical environment of the agent. Descriptions of entities (physical structure and function) and plans are stored both in the episodic memory and in the semantic memory (Aitkenhead & Slack, 1987). We will now describe in more detail each one of these distinct components.

2.1.1. Metric Map

In our approach, a (grid-based) metric map of the world is a three-dimensional grid in which a cell contains the information of the set of entities that may alternatively occupy the cell and the probability of this occupancy. Thus, each cell <*x*,*y*> of the metric map of an agent *i* is set to a set of pairs $\phi^i_{x,y}=\{<p^i_1, E^i_1>, <p^i_2, E^i_2>, ..., <p^i_{n_i}, E^i_{n_i}>, <p^i_{n_{i+1}}, 0>\}$, where E^i_j is the identifier of the j^{th} entity that may occupy the cell <*x*,*y*> of the metric map of agent *i* with probability $p^i_j \in [0,1]$, and such that $\sum_{j=1}^{n_{i+1}} p^i_j = 1$. Note that the pair $<p^i_{n_{i+1}}, 0>$ is included in order to express the probability of the cell being empty. Cells that are completely unknown, i.e., for which there are not yet no assumptions/expectations about their occupancy, are set with an empty set of pairs $\phi^i_{x,y}=\{\}$. Note also that each entity may occupy more than a single cell, i.e., there might be several adjacent cells with the same E^i_j.

2.1.2. Memory for Entities

The set of descriptions of entities perceived from the environment are stored in the *episodic memory of entities*. Each one of these descriptions is of the form <*ID,PS,F*>, where *ID* is a number that uniquely identifies the entity in the environment, *PS* is the physical structure, and *F* is the function of the entity. The sensors may provide incomplete information about an entity (for instance, only part of the physical structure may be seen or the function of the entity may be undetermined). In this case the missing information is filled in by making use of the conditional probabilistic Bayes's rule (Shafer & Pearl, 1990), i.e., the missing information is estimated taking into account the available information and descriptions of other entities previously perceived and already stored in the *episodic memory of entities*. This means some of the descriptions of entities stored in memory are uncertain or not completely known (e.g.: element 4 of Figure 2).

The physical structure of an entity may be described analogically or propositionally (Aitkenhead & Slack, 1987). The analogical representation reflects directly the real physical structure while the propositional representation is a higher level description (using propositions) of that real structure.

The analogical description of the physical structure of an entity comprises a three-dimensional matrix and the coordinates of the gravity centre relatively to the entity and to the environment spaces. Notice that the three-dimensional matrix of the entity is a submatrix of the matrix that represents the metric map.

The propositional description of the physical structure of an entity relies on the representation through semantic features or attributes much like in semantic networks or schemas (Aitkenhead & Slack, 1987). Entities are described by a set of attribute-value pairs that can be graph-based represented.

The function is simply a description of the role or category of the entity in the environment. For instance, a house, a car, a tree, etc. Like the description of the physical structure, this may be probabilistic because of the incompleteness of perception. This means, this is a set $F = \{<function_i, prob_i>: i=1,2, \ldots, n$, where n is the number of possible functions and P("function" = $function_i) = prob_i\}$.

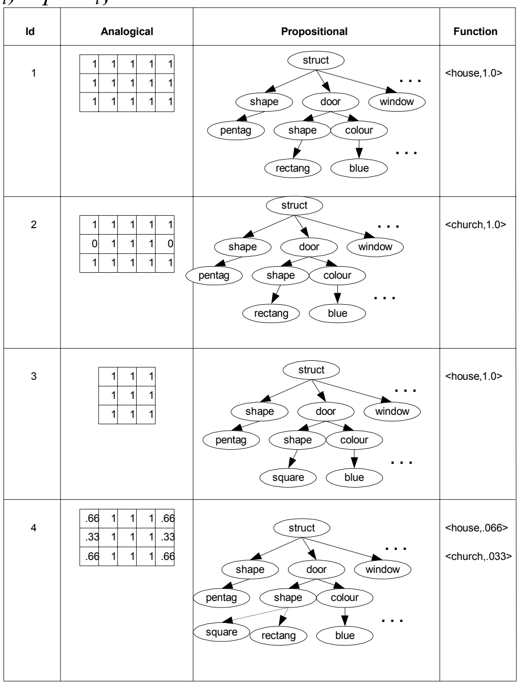

Figure 2. Example of the episodic memory of entities. Although the matrix of the analogical description is of three-dimensional kind, for the sake of simplicity, it is represented here as a two-dimensional matrix corresponding to the upper view of the entity.

Concrete entities (i.e., entities represented in the episodic memory) with similar features may be generalized or abstracted into a single one, an abstract entity, which is stored in the semantic memory for entities.

2.1.3. Memory for Plans

Like entities, we may distinguish two main kinds of plans: concrete plans, i.e., cases of plans (Kolodner, 1993), and abstract plans.

We represent plans as a hierarchy of tasks (a variant of HTNs) (e.g., (Erol, Hendler, & Nau, 1994)) (see Figure 3). Formally, a plan is a tuple *AP* = <*T*, *L*>, where *T* is the set of tasks and *L* is the set of links. This structure has the form of a planning tree, i.e., it is a kind of AND/OR tree that expresses all the possible ways to decompose an initial task network. Like in regular HTNs, this hierarchical structure of a plan comprises primitive tasks or actions (non-decomposable tasks) and non-primitive tasks (decomposable or compound tasks). Primitive tasks correspond to the leaves of the tree and are directly executable by the agent, while compound tasks denote desired changes that involve several subtasks to accomplish it. Tasks that are the roots of HTN plans are called *goal tasks*. For instance, the leaf node *PTRANS* of Figure 3 is a primitive task, while *visitEntity* is a compound task (and also a goal task).

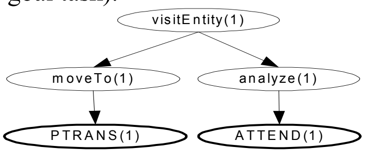

Figure 3. A simple example of plan. Primitive tasks are represented by thick ellipses while non-primitive tasks are represented by thin ellipses.

A task T is both conditional and probabilistic (e.g.: (Blythe, 1999)). This means each task has a set of conditions C={ c_1, c_2, ..., c_m} and for each one of these mutually exclusive and exhaustive conditions, c_i, there is a set of alternative effects ε^i={$<p_1^i, E_1^i>$, $<p_2^i, E_2^i>$, ..., $<p_{n_i}^i, E_{n_i}^i>$}, where E_j^i is the j^{th} effect triggered with probability $p_j^i \in [0,1]$ by condition c_i (i.e., $P(E_j^i | c_i) = p_j^i$), and such that $\sum_{j=1}^{n_i} p_j^i = 1$. Figure 4 presents the structure of a task. The probabilities of conditions are represented in that structure although we assume that conditions are independent of tasks. Thus, $P(c_i|T)=P(c_i)$. The main reason for this is to emphasize that the Expected Utility (EU) of a task, in addition to the probability of effects, depends on the probability of conditions too. In addition to conditions and effects, a task has other information components.

Each effect comprises itself a few components of several kinds such as temporal, emotional etc. These components may be of two kinds: non-procedural (factual) and procedural. The non-procedural component refers to the data collected from previous occurrences of the effect (contains the duration of the task, the emotions and respective intensities felt by the agent, the fuel consumed, etc., in previous executions of the task as stored in cases of plans). The procedural component refers to the process through which the temporal, emotional and other kinds of data may be computed (contains descriptions or rules of how to compute the components).

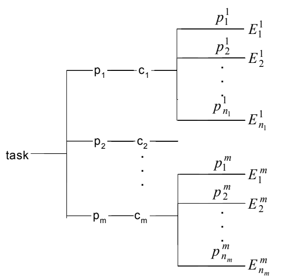

Figure 4. Schematic representation of a task in an abstract plan.

2.2 Motivations

This module receives information from the current state of the environment and outputs the intensities of motivations (emotions, drives and other motivations). In this paper, this module is confined to the motivations that are related with variables that directly influence the main activities that the agent exhibits (exploration and creativity[1]): surprise (elicited by unexpectedness), curiosity (elicited by novelty). In addition, we also consider the influence of the drive "hunger" that reflects the need of a power source. Nonetheless, other emotions, drives and other motivations may be included in this module but not considered for the purpose of this paper.

The agent is almost continuously presented with an *input proposition* (Ortony & Partridge, 1987), which corresponds to some sensorial information of an entity (for instance, "a house with squared windows"). In response to this external stimulus, the surprise and curiosity unit outputs the intensity of these motivations, respectively.

In what concerns to surprise, we have developed a computational model (Macedo & Cardoso, 2001a) with the collaboration of the psychologists of the University of Bielefeld, Germany (Meyer, Reisenzein, & Schützwohl, 1997), and also based on the ideas of Ortony and Partridge (Ortony & Partridge, 1987). The idea behind this model is that surprise consists of the appraisal of unexpectedness. Actually, there is experimental evidence supporting that the intensity of felt surprise increases monotonically, and is closely correlated with the degree of unexpectedness (see (Macedo & Cardoso, 2001a) for more details). This means that unexpectedness is the proximate cognitive appraisal cause of the surprise experience. Considering this evidence, we have already proposed (Macedo & Cardoso, 2001a) that the surprise felt by an agent *Agt* elicited by an object Obj_k is given by the degree of unexpectedness of Obj_k, considering the set of objects present in the memory of the agent *Agt*, which is given by the improbability of Obj_k (see(Macedo & Cardoso, 2001a) for more details):

$$SURPRISE(Agt, Obj_k) = \\ UNEXPECTEDNESS(Obj_k, Agt(Mem)) = 1 - P(Obj_k)$$

We define curiosity (following McDougall (McDougall, 1908), Berlyne (Berlyne, 1950) and Shand (Shand, 1914)) as the desire to know or learn an object that arouses interest by being novel, which means that novel objects stimulate actions intended to acquire knowledge about those objects. Thus, if we accept the above definition, the curiosity induced in an agent *Agt* by an object Obj_k depends on the novelty or difference of Obj_k relatively to the set of objects present in the memory of *Agt*:

$$CURIOSITY(Agt, Obj_k) = DIFFERENCE(Obj_k, Agt(Mem))$$

[1] The agents that we have implemented have been used to explore unknown environments (Macedo & Cardoso, 2001b), and to create things (Macedo & Cardoso, 2001c).

The measure of difference relies heavily on error correcting code theory (Hamming, 1950): the function computes the distance between two objects represented by graphs, counting the minimal number of changes (insertions and deletions of nodes and edges) required to transform one graph into another.

The drive hunger is defined as the need of a source of energy. Given the capacity C of the storage of that source in an agent, and L the amount of energy left ($L \leq C$), the hunger elicited in an agent is computed as follows:

$HUNGER(Agt)=C-L$

2.3 Goals/Intentions and Desires

Desires are states of the environment the agent would like to happen, i.e., they correspond to those states of the environment the agent prefers. This preference is implicitly represented in a mathematical function that evaluates states of the environment in terms of the positive and negative feelings they elicit in the agent. This function obeys to the Maximum Expected Utility (MEU) principle (Russel & Norvig, 1995). The agent prefers always those states that make it feel more positive feelings (more positive emotions and the satisfaction of drives). Goals or intentions may be understood as something that an agent wants or has to do. These might be automatically generated by the agent or given by other agents.

2.4 Deliberative Reasoning/Decision-making

The reasoning and decision-making module receives information from the internal/external world and outputs an action that has been selected for execution. Roughly speaking, the agent starts by computing the current world state. This is performed taking into account the information provided by the sensors (which may be incomplete) and generating expectations or assumptions for the missing information. Assumptions and expectations for the current agent's position are also generated. The agent has a queue of goal tasks/intentions ranked by their priority (i.e., EU). The first of the ranking is the goal/intention that is under achievement. Once one goal is achieved, it is removed from the queue and the way it was achieved could be learned for future reuse by simply storing its plan in memory as a case. However, external events or objects, for instance, may give rise to new goals/intentions. This is the next step of the reasoning/decision-making process: the generation of new intentions/goals, computation of their EU and insertion of them in the queue of goals/intentions according to their priority (i.e., their EU). Though, if the queue was empty before this step and no new goals are generated in this step, the queue remains empty. In this case there is nothing to reasoning or deciding about and consequently no action is returned. However, the most likely is that the queue is not empty either before or after the step of generating new goals. If the first goal of the queue is still the same then proceed with its execution and possibly replanning if necessary. However, the addition of new goals may have caused changes in the ranking of the goals in the queue because a new goal may be more EU than some old goals. Thus, the first goal may now be different from the previous first goal. In this case the old first goal is considered suspended. This suspension could happen even though the goal was already under achievement (there was already a plan built for it and this plan was already being executed). Thus, a plan is required for this new first goal in queue, which will be from now on the current goal until its achievement or suspension. That plan could be built or retrieved from memory (if there is one – remember that this current goal may be previously suspended or even previously achieved in the past).

The generation of plans is performed much like in HTN approaches (see (Erol et al., 1994)). We will now describe in more detail the step related with the generation and ranking of agent's goals.

2.4.1. Generation and Ranking of Goals/Intentions

The motivational system plays an important role in the generation and ranking of goals/intentions. Actually, according to psychologists, motivations are the source of goals in several manners: these goals may be included in emotions (e.g., when an agent feels anger about something, a possible triggered goal might be fisting the entity that is on the origin of the anger), or emotions may be themselves the goals (e.g., an agent looks for states of the world that elicit certain positive emotions such as happiness or surprise). Therefore, an agent selects actions or sequences of actions that lead to those states of the world. For instance, an agent establishes the goal of visiting an object that seems beforehand interesting (novel, surprising) because visiting it will probably make it feel happy. The algorithm for the generation and ranking of goals/intentions is as follows (see Figure 5). First, the set of different goal tasks present in the memory of plans are retrieved and, for each kind, a set of new goals (*newGoals*) is generated using the function *adaptGoal*(). This function takes as input a goal task retrieved from a plan in the memory of plans, the memory and the perception of the agent, and generates similar goals resulting from the adaptation of the past goal to situations of the present state of the world. The adaptation strategies used are mainly substitutions (Kolodner, 1993). Thus, for instance, suppose the goal task *visitEntity*(*e7*) is present in the memory of the agent. Suppose also that the agent has just perceived three entities present in the environment, *e1*, *e2* and *e3*. The entity to which *visitEntity* is applied (*e7*) may be substituted by *e1*, *e2* or *e3*, resulting three new goals: *visitEntity*(*e1*), *visitEntity*(*e2*), *visitEntity*(*e3*). Then, the EU of each goal task is computed. As said above, a task T is both conditional and probabilistic (e.g.: (Blythe, 1999)). Thus, the execution of a goal task under a given condition may be seen according to Utility Theory as a lottery (Russel & Norvig, 1995):

$$Lottery(T) = \left[p^1 \times p_1^1, E_1^1; p^1 \times p_2^1, E_2^1; ...; p^m \times p_{n_m}^m, E_{n_m}^m \right]$$

, where p^i is the probability of the condition c_i, p_j^i is the probability of the j^{th} effect, E_j^i, of condition c_i.

The EU of T may be then computed as follows:

$$EU(T) = \sum_{k,j} p^k \times p_j^k \times EU(E_j^k)$$

The computation of $EU(E_j^k)$ is performed predicting the emotions that could be elicited by achieving/executing the goal task. This means, the emotions, drives and other motivations felt by the agent when the effect takes place are predicted or estimated based on the procedural or non-procedural components of the effect.

The following function is used to compute $EU(E_j^k)$:

$$EU(E_j^k) =$$

$$= \frac{\alpha_1 \times U_{surprise}(E_j^k) + \alpha_2 \times U_{curiosity}(E_j^k) + \alpha_2 \times U_{hunger}(E_j^k)}{\sum_i \alpha_i} =$$

$$= \frac{\alpha_1 \times SURPRISE(E_j^k) + \alpha_2 \times CURIOSITY(E_j^k) + \alpha_2 \times HUNGER(E_j^k)}{\sum_i \alpha_i}$$

, where, α_2 = -1 and α_i ($i \neq 2$) may be defined as follows:

$$\alpha_i = \begin{cases} 1 \Leftarrow C - HUNGER(Agt) - D > 0 \\ 0 \Leftarrow otherwise \end{cases}$$

, where D is the amount of energy necessary to go from the end location of goal task T to the closer place where energy could be recharged, and C is the maximum amount of energy that could be stored by the agent.

In the case of exploratory and creativity behaviour, the surprise and curiosity of an effect of a task are elicited by the objects that the agent perceives.

Algorithm generateRankGoals(*newRankedGoals*)
Output: *newRankedGoals* – the set of ranked goals

newGoals ← ∅
setPastGoals ← {*x*: *x* is a goal task belonging to some plan in memory}
for each *goal* **in** *setPastGoals* **do**
    adaptationGoal←adaptGoal(*goal*,*agtMemy*,*agtPercepts*)
    newGoals ← *newGoals* ∪ *adaptationGoals*
end for each
for each *goal* **in** *newGoals* **do**

$$EU(T) = \sum_{k,j} p^k \times p_j^k \times EU(E_j^k)$$

end for each
insert(*goal*,*newRankedGoals*)
return *newRankedGoals*
end

Figure 5. Algorithm for the generation and ranking of goals.

3 Qualitative Experiment

We have conducted an experiment in order to evaluate the reasoning/decision-making process of an agent with the architecture described above. Special attention was given to the algorithm of the autonomous generation and ranking of goals based on motivations. To do so, we ran an agent in a simulated environment populated with several buildings (their functions were for instance, house, church, hotel, etc.; for the sake of simplicity, their descriptions were related with the shapes of their structure: rectangular, squared, etc.). Figure 6 presents the simulated environment and the path taken by the agent to explore it. The agent started at location 0, with an empty memory of entities, but with a single case of a past plan for visiting entities. At this location its visual field included objects *E1* and *E2*, located respectively at locations 1 and 2. Then the agent generated goals for visiting them by adapting the goal *visitEntity* of the previous plan stored in memory. The resulting goals are: *visitEntity(E1)* and *visitEntity(E2)*. *E1* and *E2* are entirely new for the agent (remember that the agent started with an empty memory of entities). Therefore, the surprise and curiosity that they may elicit when visited is maximum (i.e., 1.0). However, *E1* is closer, so the hunger that may be felt when the agent is at location 1 is lower than in location 2. Hence, the agent ranks the goals as follows: *visitEntity(E1)* followed by *visitEntity(E2)*. A plan is generated for the first goal. After its execution, the agent is at location 1 with a complete description of *E1* stored in memory as a case (case 1 of the episodic memory of Figure 2) and an incomplete description of *E2* (because it has not been visited yet and therefore it is not completely known – at least the function is still undetermined). In addition, the goal *visitEntity(E1)* is deleted from the queue of goals. At location 1, the agent perceives *E2* and *E3* (*E1* is also perceived, but it has just been visited). The agent generates the goal *visitEntity(E3)* for visiting *E3*. Notice that *visitEntity(E2)* is still in the queue of goals. *E3* is similar to the previously visited *E1* and therefore it predicts feeling a low intensity of surprise and curiosity when visiting it. Besides, hunger is expected to be higher in location 3 than in 2. So, the goals are ranked as follows: *visitEntity(E2)* followed by *visitEntity(E3)*. Once again, a plan is generated for *visitEntity(E2)* and then executed. The result is the completion of the description of *E2* (case 2 of the episodic memory of Figure 2). At location 2, the agent perceives *E4*, in addition to *E3*. *E4* is similar to both *E1* and *E2*. However, its EU is lower than that of *E3* mainly because the agent expects a higher hunger in location 4 than in 3. Thus, *E3* is visited. At this time, the agent has the episodic memory of Figure 2. An interesting behaviour is observed later when the agent has to select between visiting *E11* and *E12*, which are exactly equal to *E1* and *E2*, respectively, and at similar distances. Therefore, it might be expected that the agent would visit *E11*. However, this time the agent ranks the goals as follows: *visitEntity(E12)* and *visitEntity(E11)*. This is because the agent has now more cases describing entities similar to *E11* than to *E12*. Therefore, *E12* is expected to elicit more surprise than E11, and hence the EU of visiting *E12* is higher than that of visiting *E11*.

In order to take conclusions about the quality of this behaviour, we asked a few humans to describe the path they would follow in such environment. We verified that there is much similarity with the path followed by the agent.

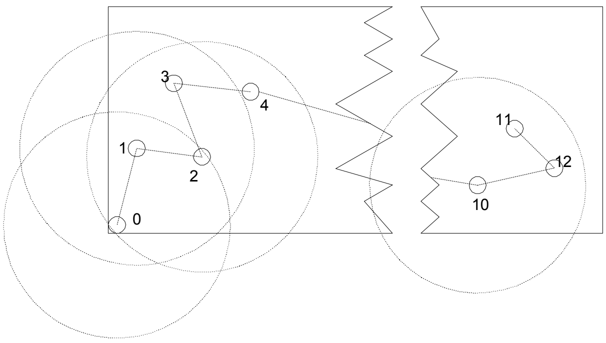

Figure 6. Experiment in a simulated environment. Dashed circles represent the visual field of the agent in different locations.

4 Conclusions

We have presented a motivation-based approach for the autonomous generation and ranking of goals. This approach is in the core of the reasoning process of agents. The experiment conducted allows us to conclude that the behaviour of an agent whose reasoning process includes this approach is similar to that of humans in the simulated environment considered.

Aknowledgements

The PhD of Luís Macedo is financially supported by PRODEP III.

References

Aitkenhead, A., & Slack, J. (1987). *Issues in cognitive modelling*. London: Lawrence Erlbaum Associates.

Berlyne, D. (1950). Novelty and curiosity as determinants of exploratory behavior. *British Journal of Psychology, 41*, 68-80.

Blythe, J. (1999). Decision-Theoretic Planning. *AI Magazine, Summer 1999*.

Damásio, A. (1994). *Descartes'error, Emotion Reason and the Human Brain*. New York: Grosset/Putnam Books.

Erol, K., Hendler, J., & Nau, D. (1994). UMCP: A sound and complete procedure for hierarchical task-network planning, *Proceedings of the International Conference on AI Planning Systems* (pp. 249-254).

Etzioni, O., & Weld, D. (1995). Intelligent agents on the internet: fact, fiction, and forecast. *IEEE Expert, 10*(44-49).

Hamming, R. (1950). Error Detecting and Error Correcting Codes. *The Bell System Technical Journal, 26*(2), 147-160.

Izard, C. (1991). *The Psychology of Emotions*. NY: Plenum Press.

Kolodner, J. (1993). *Case-Based Reasoning*. San Mateo, CA: Morgan-Kaufmann.

Macedo, L., & Cardoso, A. (2001a). Modelling Forms of Surprise in an Artificial Agent. In J. Moore & K. Stenning (Eds.), *Proceedings of the 23rd Annual Conference of the Cognitive Science Society* (pp. 588-593). Mahwah, NJ: Erlbaum.

Macedo, L., & Cardoso, A. (2001b). SC-EUNE - Surprise/Curiosity-based Exploration of Uncertain and Unknown Environments., *Proceedings of the AISB'01 Symposium on Emotion, Cognition and Affective Computing* (pp. 73-81). York, UK: University of York.

Macedo, L., & Cardoso, A. (2001c). Using Surprise to Create Products that get the Attention of other Agents. In L. Canãmero (Ed.), *Emotional and Intelligent II: The Tangled Knot of Social Cognition - Papers from the 2001 AAAI Fall Symposium (Technical Report FS-01-02)* (pp. 79-84). Menlo Park, CA: The AAAI Press.

McDougall, W. (1908). *An introduction to social psychology*. London: Methuen.

Meyer, W., Reisenzein, R., & Schützwohl, A. (1997). Towards a process analysis of emotions: The case of surprise. *Motivation and Emotion, 21*, 251-274.

Ortony, A., & Partridge, D. (1987). Surprisingness and Expectation Failure: What's the Difference?, *Proceedings of the 10th International Joint Conference on Artificial Intelligence* (pp. 106-108). Los Altos, CA: Morgan Kaufmann.

Rao, A., & Georgeff, M. (1995). BDI Agents: From Theory to Practice, *Proceedings of the First International Conference on Multiagent Systems*.

Reisenzein, R. (1996). Emotional Action Generation. In W. Battmann & S. Dutke (Eds.), *Processes of the molar regulation of behavior*. Lengerich: Pabst Science Publishers.

Russel, S., & Norvig, P. (1995). *Artificial Intelligence - A Modern Approach*. Englewood Cliffs, NJ: Prentice Hall.

Shafer, G., & Pearl, J. (Eds.). (1990). *Readings in Uncertain Reasoning*. Palo Alto, CA: Morgan Kaufmann.

Shand, A. (1914). *The foundations of character*. London: Macmillan.

Thrun, S. (2002). Robotic mapping: A survey. In G. Lakemeyer & B. Nebel (Eds.), *Exploring Artificial Intelligence in the New Millenium*. San Mateo, CA: Morgan Kaufmann.