Applying Appraisal Theories to Goal Directed Autonomy

Robert P. Marinier III, Michael van Lent, Randolph M. Jones

Soar Technology, Inc. 3600 Green Court, Suite 600, Ann Arbor, MI 48105
{bob.marinier,vanlent,rjones}@soartech.com

Abstract

Appraisal theories (Roseman and Smith, 2001) are psychological theories exploring how humans evaluate situations along a number of appraisal dimensions, many of which compare the situation to the current goals. We report on the application of appraisal theories to focus the learning of policies in complex domains via Soar 9's built-in reinforcement learning mechanism. In addition, we describe how appraisal theories relate to and might be applied to Goal Directed Autonomy.

Introduction

Goal Directed Autonomy (Molineau, Klenk, and Aha 2010) is designed to address complex environments that are non-deterministic, dynamic and continuous, and for tasks whose goals can be partially satisfied and dynamically changed. The GDA approach combines four features to address these issues: discrepancy detection, discrepancy explanation, goal generation, and goal management.

Numerous models have been created in the Soar cognitive architecture (Newell, 1990; Laird, 2008) that take various approaches to each of these problems. In this paper, we will focus on the application of appraisal theories (Roseman and Smith 2001) in Soar to address these problems. We will begin by describing previous work in this area focused on learning policies. These policies are similar to the System in the GDA conceptual model but include learned preferences for which action to take in each state. Thus, the learned policy implicitly encodes a plan eliminating the need for a Planner. The role of the Controller (in this case executing the policy) is filled by the Soar architecture. Applying appraisal theories to learn policies highlights an alternative approach to performance in complex domains. In the second half of the paper we describe ideas for using appraisal theories to more directly address the four features of GDA. For example, appraisals are directly applicable to discrepancy detection and the system described here even uses "Discrepancy from Expectation" as one of the three appraisal dimensions.

Background

Soar

The Soar cognitive architecture (Newell 1990) is a domain- and task-independent agent framework. Soar is composed of a small set of general purpose memories and processes which combine to create complex behaviors. By itself Soar does nothing; however, by adding knowledge to its long-term memories and placing it in an environment, Soar comes to life, retrieving long-term knowledge applicable to its perceptions and internal state representation to generate behaviors and solve problems. That is, Soar makes no commitment to particular problem-solving approaches (e.g., planning); rather, the approaches used are determined by the agent's knowledge (e.g., the agent may have knowledge about how to plan). The knowledge added to Soar is typically a mix of general knowledge and domain-specific procedural knowledge. Additional knowledge can be learned through experience, resulting in improved performance over time. Finally, there is no limit to the kinds or amount of knowledge that a Soar agent can have; thus, a single agent could have knowledge of how to operate across many domains and tasks.

Originally Soar was composed of a single associative long-term procedural memory and a single learning mechanism for learning new procedural knowledge. However, work in the last several years has resulted in Soar 9, a new version of the Soar architecture that includes several new memories and learning mechanisms (Laird 2008), including episodic and semantic memories and associated learning mechanisms, and the addition of reinforcement learning to tweak procedural knowledge.

Appraisal Theories

Appraisal theories (see Roseman and Smith, 2001 for an overview) are psychological theories of the antecedents of emotion. While there are several competing versions, the

basic claim is that humans evaluate situations along a number of appraisal dimensions, many of which compare the situation to the current goals. Example dimensions are shown in Table 1. Patterns of appraisal values map to specific emotional reactions; for example, high Goal Relevance, low Goal Conduciveness, Agency of "other", and high Control may result in Anger (someone caused something bad to happen, and you can do something about it). On the other hand, the same situation with low Control may result in Fear (someone caused something bad to happen, and you cannot do anything about it). Other common emotions include Joy, Sadness, Guilt, and Pride.

Appraisal Dimension	**Description**
Novelty	Uniqueness of the situation
Goal Relevance	How important the situation is for my goals
Goal Conduciveness	How good or bad the situation is for my goals
Agency	Who caused the situation and why
Outcome Probability	How likely a predicted outcome is
Discrepancy from Expectation	How different an outcome is from my prediction
Urgency	How important is it to act immediately
Control	To what extent can I change the situation

Table 1: Example appraisal dimensions.

While the exact set of appraisal dimensions, emotions, and the mapping between them varies between theories, for our purposes the key feature is that there is a domain and task independent description of situations that can be used to support behavior generation, learning, and many other processes.

Intrinsically Motivated Reinforcement Learning

In traditional reinforcement learning (Sutton and Barto, 1998), agents receive an external task-specific reward signal indicating the desirability of the current situation for some fixed goal. This reward signal is used to learn which actions to perform in which states, such that, after much experience, the reward signal is maximized, resulting in optimal task performance.

Intrinsically motivated reinforcement learning (Singh, Barto, and Chentanez, 2004) is the idea that an agent can generate its own internal reward signal, and thus motivate its own learning. In our work described below, we explored using appraisals to generate the reward signal for Soar's reinforcement learning mechanism. That is, rather than mapping appraisal values to specific emotions, we map them to rewards. Reward in this case could be viewed as a very general "good/bad" emotion. This approach introduces an element of domain- and task-independence to reward generation. The generation of specific appraisal values requires some domain- and task-specific knowledge, but the mapping from appraisal values to rewards is domain- and task-independent. Furthermore, this enables an agent to dynamically switch goals without necessarily knowing ahead of time what those goals might be. That is, since the agent controls the reward function, it can adapt to changing and unpredictable situations more readily than an externally fixed reward function.

Previous Work

Our research into adapting appraisal theories to Soar builds on a mature body of work developing intelligent agents that fluidly manage goals to generate autonomous behavior. There is a family of knowledge-rich, interactive intelligent agents developed in Soar, of which the most sophisticated example is TacAir-Soar (Jones et al. 1999). TacAir-Soar is a model of fixed-wing pilots flying military missions. TacAir-Soar operates in a non-deterministic, dynamic, continuous environment with numerous goals that sometimes interact and that cannot always be achieved (Jones et al, 1994). In contrast to the explicit planning approach of GDA, TacAir-Soar implements a form of reactive planning, enhanced with significant reasoning capabilities for understanding the wide variety of situations a military pilot may encounter. TacAir-Soar encodes into its procedural knowledge methods for interpreting and understanding changes in the environment and translating changes into higher-level beliefs about the current (observable and unobservable) state of the world. Additional procedural knowledge encodes the reasoning required to generate behavior appropriate to the interpreted situation. Behavior generation is accomplished by first adaptively activating (and deactivating) sets of hierarchical goals. The activation, monitoring, and reconsideration of goals are all determined by procedural knowledge. Because TacAir-Soar does not create or maintain explicit declarative plans (beyond a high-level mission description), it also does not create explicit, declarative predictions about how the plans should unfold. Instead, predictions about the results of agent actions are implicit in the procedural knowledge for monitoring and reconsidering goals. Discrepancy detection amounts to detecting when goal-activation conditions are violated, resulting in the automatic reconsideration of the goals, as well as an goal entailments that have been created by the procedural knowledge. Updated situation representations then trigger other goal-activating procedural knowledge, providing a new goal/situation context for behavior. All subsequent behavior is generated in the combined context of the interpreted situation and the active goal set, including internal behaviors such as further goal generation and maintenance, and further situation interpretations, as well as external behaviors representing interactions with the environment. Explicit discrepancy explanation is unnecessary in this approach, as the agent has knowledge of how to behave in all situations; that is, the explanations are implicit in the procedural knowledge. The requirement that the system know how to behave in all situations may appear onerous; however, this is managed by maintaining twin goal and situation representation hierarchies. Default

behaviors ensure responses for higher-level goals and situation interpretations. However, ensuring that this does not result in bad behavior still requires virtually complete knowledge of the "interesting" combinations of goals and situations that the agent will encounter.

One of the features of TacAir-Soar (and related systems) that sets it apart from earlier reactive planning systems is the careful attention paid to creating rich representations of situational understanding. Much of TacAir-Soar's knowledge base is devoted to making inferences about the state of the world given primitive sensory data. These situation interpretations can be thought of as domain-specific forms of situation appraisal.

The ARTUE system (Molineau, Klenk, and Aha 2010) employs an HTN planner to generate explicit plans and predictions. Discrepancies are detected by comparing these predictions to observed states. Explanations are generated by proposing possible assumptions that could be made about unobservable parts of the state, and determining which of these can explain the discrepancy. Validated assumptions are then added to the set of beliefs, which can trigger the generation of a new goal. Goal generation is supported by schemas that describe patterns of beliefs that trigger the generation of new goals (similar to Soar's procedural memory). Goal management is supported by the intensity property of goals. Plans are generated to achieve the highest intensity goal. In comparison to TacAir-Soar, which captures all required knowledge in procedural memory, ARTUE employs multiple knowledge types including declarative representations of domain knowledge to generate plans, schemas to generate goals, and specialized processes to detect discrepancies and manage goals. Similar to TacAir-Soar, however, it requires domain knowledge. Also similar to TacAir-Soar, anything it does not understand is simply ignored.

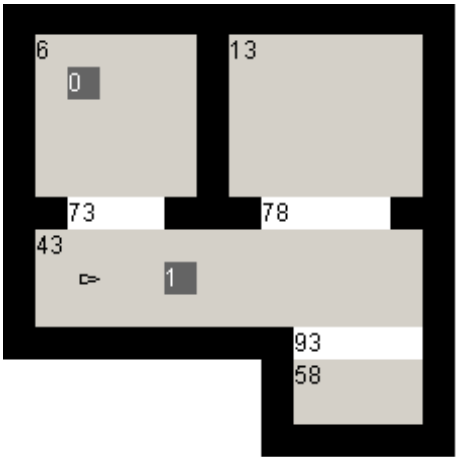

Figure 1: The Rooms World environment. Blocks are dark boxes. The agent is the triangle. Room 13 is the storage room.

Using Appraisals to Learn Policies

The GDA approach uses domain knowledge to address complex environments. However, there is no attempt to determine where this knowledge comes from. Several approaches exist to learning explicit domain knowledge (e.g., Pearson and Laird, 2005; van Lent and Laird, 2001). In this work, however, we explore learning to behave without explicit domain knowledge. In short, we use reinforcement learning to learn a policy of which actions to take in which states, without ever learning the explicit effects of those actions. Normally reinforcement learning would not work well in these kinds of complex environments, requiring thousands of episodes to converge, if convergence is achieved at all. This is because typical approaches would only provide a reward signal when the task is completed. This signal is diluted as it propagates backwards to earlier states. In complex environments, this results in many states never getting strong values, and thus learning is nearly impossible. In our approach, since the agent is able to generate frequent reward feedback via appraisals during the task, learning occurs very rapidly.

Rooms World

The domain is called Rooms World. Rooms World is composed of a set of connected rooms, some of which contain blocks. One of the rooms is designated the storage room. The agent's task is to collect the blocks and move them to the storage room. The agent can turn and move forward, and pick up and put down a block. The agent can only carry one block at a time. The agent's movement is continuous and takes time (it turns and moves at a fixed rate). The agent's perception is limited by a vision cone.

The agent's task is divided into three goals. The top-level goal is to Clean-House. There are two subgoals, one to Goto-Room(x), and another to Clean-Current-Room. Cleaning the current room means removing all blocks from it. The agent has internal actions that can create a subgoal, retrieve the supergoal, and mark a goal as completed.

The agent has knowledge of when it can do each of the possible actions (both external and internal), but does not have knowledge of their effects. The agent also does not initially know the layout of the world, how many rooms or blocks there are, or where the storage room is (although it does know how to recognize the storage room).

As the agent behaves in the world, it builds a map of the world and also a task model that relates actions in particular states to the next-perceived stimulus (which may be internal or external). The agent has an attention mechanism that restricts it to processing to one (internal or external) feature at a time. Thus, the next-perceived stimulus is non-deterministic, resulting in a task model that captures regularities.

The underlying Rooms World model is deterministic, but the partial observability and attention model make it appear non-deterministic to the agent. The world is only dynamic with respect to the agent's actions (i.e., when the

agent moves blocks around), but its lack of knowledge about the initial state of the world makes it appear dynamic to the agent (additional rooms and blocks "appear" to the agent as it explores). Rooms World is continuous, and the agent can dynamically change its goals.

Appraisals and Learning in Rooms World

For each stimulus and action, we generated values for three appraisal dimensions: Goal Conduciveness, Outcome Probability, and Discrepancy from Expectation.

Goal Conduciveness was generated differently for different goals. For the Clean-House goal, Conduciveness was generated by tracking the number of known blocks that need to be moved to the storage room over time. If this number is going down, then Conduciveness is high; otherwise it is low. For the Goto-Room(x) goal, the distance (in rooms) to the target room is tracked over time; if the distance is decreasing, then Conduciveness is high; otherwise it is low. For Clean-Current-Room, if the agent removes a block from the room, Conduciveness is high; otherwise it is low.

Outcome Probability is generated from the task model. For a given state, the occurrences of all observed next stimuli are compared, and the most common one is predicted to occur. The Outcome Probability of this prediction is determined by how much more common it is than the other possibilities.

Discrepancy from Expectation was generated as a binary value; if the prediction exactly matched the observed stimulus, then Discrepancy was low; otherwise, Discrepancy was high.

The conversion of these values to a reward signal is beyond the scope of this paper; the interested reader is directed elsewhere (Marinier and Laird, 2007; Marinier 2008).

Results

Results shown in Figure 2are the medians of 50 trials of 15 identical episodes each. Episodes are depicted on the horizontal axis. The left vertical axis shows time (measured in Soar decision cycles) while the right vertical axis shows total reward accumulated during an episode. Optimal Time is difficult to determine exactly, but an approximation is shown. There is no optimal Total Reward since the conversion from appraisals to reward makes the reward signal non-stationary (see Marinier 2008 for details). The agent approaches near-optimal behavior in about 10 episodes. This extremely rapid learning results from frequent reward feedback. Whereas a standard reinforcement learning agent only gets a reward signal upon completing the task, our agent generates its own rewards and thus gets frequent feedback during the task. Additionally, the ability to divide the task into subtasks speeds learning.

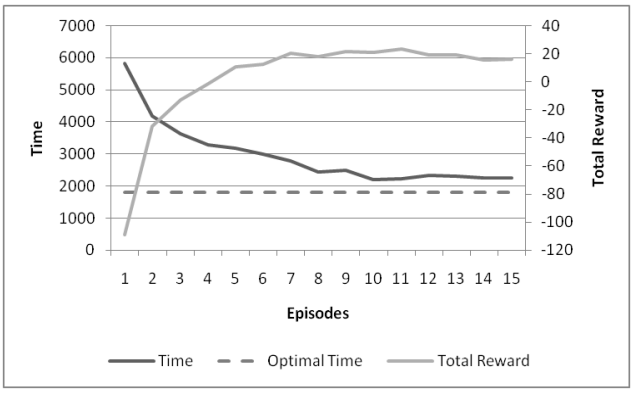

Figure 2: Time to completion and total reward accumulated as learning occurs across 15 episodes.

Appraisals for GDA

The above system has some aspects of GDA but does not address others. It does not perform explicit planning; rather, "plans" are implicit in the policy it is learning. This is a feature of Soar: no specific method for generating behavior is "hardcoded" in the architecture – rather, knowledge combined with the current situation determines the approach used. Thus, knowledge could result in HTN planning, or reactive planning, or anything in between. Our agent has knowledge of legal actions, but not the task knowledge of when to use each action.

Discrepancy detection is directly supported by appraisal theories in the form of the Discrepancy from Expectation appraisal. Appraisal theories generally do not make claims about how the level of discrepancy is computed. Our approach was very simple and turned out to be adequate, but a more sophisticated approach like that used by ARTUE has the potential to improve performance by generating more fine-grained distinctions.

In order to perform discrepancy detection, however, some kind of prediction must be generated. While our agent does not have explicit plans, it uses its task model to generate explicit predictions. These predictions are about the next stimulus the agent will attend to, but in general, predictions could be about anything. For example, an agent could predict abstract, domain- and task-independent features like progress toward the goal (Marinier and Laird 2008). In addition to generating a prediction, the agent also generates the Outcome Probability appraisal, which describes how likely the agent thinks the prediction is to be true. This is useful in cases where the agent is unsure about what will happen next (common in the early stages of learning). That is, if Discrepancy is high but Outcome Probability is low, then that has a qualitatively different meaning than if Outcome Probability were high in that case.

Discrepancy explanation is not performed by our agent; indeed, it is not clear that it is actually necessary to perform in complex environments especially if the environment does not change in unexpected ways (as demonstrated by our agent). Still, discrepancy explanation is likely to enhance performance when it can be performed.

Agency (who caused the situation and why) describes key aspects of discrepancy explanation (but can also be applied when there is no discrepancy). Again, appraisal theories generally do not specify how Agency would be generated. ARTUE's approach is one possibility. Another would be to use schemas to that describe situations – the observable aspects are used to match a schema, which provides information on the unobservable aspects (Taylor et al. 2007).

Goal generation and goal management are performed by our agent as internal actions; they are appraised just like external actions. That is, they are evaluated for their impact on achieving the goal. In general, several other appraisals could be brought to bear in the evaluation of goal-related actions, including Urgency and Control. Urgency is similar to ARTUE's intensity, while Control would indicate which goals the agent thinks it can actually achieve. Again, multiple processes could be used to generate these values, including ARTUE's schema approach or planning (to determine how and how quickly a goal could be achieved).

Conclusion

While appraisal theories originated in psychological studies of human emotion, they have resulted in a general way of representing and thinking about situations that can be used to support many kinds of processing, including those proposed by GDA. Combined with learning, they can also reduce up-front knowledge requirements. Future work will involve extending our simple agent to include more appraisals. Ultimately, we envision extending Soar to make appraisals, and perhaps some aspects of their generation, part of the architecture, thus making it easy to incorporate them in all future Soar agents. This may result in a shift in agent design, where this appraisal knowledge, previously encoded in task- and domain-specific representations, is represented in a general way. This may increase knowledge reuse and agent robustness.

References

Jones, R., Laird, J., Nielsen, P., Coulter, K., Kenny, P., and Koss, F. (1999). Automated Intelligent Pilots for Combat Flight Simulation. *AI Magazine*. 20(1).

Jones, R. M., Laird, J. E., Tambe, M., & Rosenbloom, P. S. (1994). Generating behavior in response to interacting goals. Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation, 317–324. Orlando, FL.

Laird, J. 2008. Extending the Soar Cognitive Architecture. In proceedings of the First Conference on Artificial General Intelligence. Memphis, Tennessee.

Marinier, R. 2008. A Computational Unification of Cognitive Control, Emotion, and Learning. Ph.D. diss., Department of Computer Science and Engineering, University of Michigan, Ann Arbor, Michigan.

Marinier, R. and Laird, J. 2007. Computational Modeling or Mood and Feeling from Emotion. In proceedings of the 29th Annual Conference of the Cognitive Science Society. Nashville, Tennessee.

Marinier, R. and Laird, J. 2008. Emotion-Driven Reinforcement Learning. In proceedings of the 30th Annual Conference of the Cognitive Science Society. Washington, D.C.

Molineaux, M., Klenk, M., and Aha, D. 2010. Goal-Driven Autonomy in Navy Strategy Simulation. To appear in Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence. Atlanta, GA: AAAI Press.

Newell, A. 1990. Unified Theories of Cognition. Cambridge, Mass.: Harvard University Press.

Pearson, D.J., Laird, J.E. Incremental Learning of Procedural Planning Knowledge in Challenging Environments. *Computational Intelligence* 21(4): 414-439.

Roseman, I. & Smith, C. A. (2001). Appraisal theory: Overview, Assumptions, Varieties, Controversies. In K. R. Scherer, A. Schorr, & T. Johnstone (Eds.) *Appraisal processes in emotion: Theory, Methods, Research* (pp. 3-19). New York and Oxford: Oxford University Press.

Taylor, G., Quist, M., Furtwrangler, S., Knudsen, K. (2007). Toward a Hybrid Cultural Cognitive Architecture. First International Workshop on Cognition and Culture at CogSci 2007, Nashville, TN.

van Lent, M. C. & Laird, J. E. (2001). Learning Procedural Knowledge through Observation. Proceedings of the First International Conference on Knowledge Capture, Victoria, B.C., Canada.