Breadth of Approaches to Goal Reasoning: A Research Survey

Swaroop Vattam SWAROOP.VATTAM.CTR.IN@NRL.NAVY.MIL
Naval Research Laboratory, NRC Postdoctoral Fellow, Washington, DC 20375

Matthew Klenk KLENK@PARC.COM
Palo Alto Research Center, Embedded Reasoning Area, Palo Alto, CA 94304

Matthew Molineaux MATTHEW.MOLINEAUX@KNEXUSRESEARCH.COM
Knexus Research Corporation, Springfield, VA 22150375

David W. Aha DAVID.AHA@NRL.NAVY.MIL
Naval Research Laboratory, Navy Center for Applied Research in AI, Washington, DC 20375

Abstract

Goal-directed behavior is a hallmark of intelligence. While the majority of artificial intelligence research assumes goals are static and externally provided, many real-world applications involve unanticipated changes in the environment that may require changes to the goals themselves. Goal reasoning, which emphasizes the explicit representation of goals, their automatic formulation and dynamic management, is considered an important aspect of high-level autonomy. Building from these three basic requirements, we describe and apply a framework for surveying research related to goal reasoning that focuses on triggers and methods for goal formulation and goal management. We also summarize current research and highlight potential areas of future work.

1. Introduction

It is generally acknowledged that goal-directed behavior is a hallmark of intelligence (Newell & Simon 1972; Schank & Abelson 1977). Goal-directed behavior has usually been interpreted as *autonomy of actions* - an intelligent agent should be able to reason about actions in an autonomous manner in order to change the state of the world (including itself) as a means to satisfying a *given* goal. On the one hand, this interpretation has provided a clear focus, guiding much AI research from early problem solvers to modern day automated planners. On the other hand, it has also limited the reach and richness of AI systems by ignoring *goals*; it is often assumed that an external user or system is responsible for providing goals that remain static over a problem-solving episode. *Goal reasoning* (e.g., Norman & Long, 1996; Cox, 2007; Hawes, 2011; Klenk, Molineaux, & Aha, 2013; Jaidee, Muñoz-Avila, & Aha, 2013) challenges this interpretation and strives for *autonomy of goals* – in addition to autonomy of actions, an intelligent agent should be aware of its own goals and deliberate upon them. As we start to consider designs for intelligent systems that are more autonomous and use multiple interacting competencies to solve a wider variety of problems in the real world, it becomes increasingly difficult to ignore the issue of goal reasoning.

To illustrate the importance of goal reasoning for intelligent behavior, consider a fishing craft in the Gulf of Mexico. While carrying out a plan to achieve the goal of catching fish, the fishermen receive reports of an explosion on a nearby offshore oil rig. Upon hearing the reports, the fishermen change their goal from “catch fish” to “rescue the rig’s workers”. This goal change results in a far superior outcome, rescued workers, but is outside the scope of the original mission, catching fish.

In this paper, we present a preliminary analysis of research related to goal reasoning in the context of planning and problem-solving. (Due to space limitations, we do not also examine research on the role of goals in human and machine learning (e.g., Leake 1991; Leake & Ram 1995).) We begin by describing the Goal Reasoning Analysis Framework (GRAF) and use it to focus on the tasks of goal formulation and goal management. Next we survey approaches and techniques for these tasks in terms of this framework. Finally, we briefly discuss current goal reasoning research and highlight potential areas for future work.

2. Goal Reasoning Analysis Framework (GRAF)

Because the notion of *goal reasoning* is polymorphous and often interpreted and applied differently in different research contexts, it is productive to think about a common framework for analyzing and comparing the various techniques and approaches related to goal reasoning. We propose the Goal Reasoning Analysis Framework (GRAF) as a first step in this direction. We develop this framework by first identifying the following three minimum requirements for goal reasoning.

Explicit goals: First, the system should explicitly represent and reason about goals.

Goal formulation: Second, the system should be able to formulate *goals*. Once we require an intelligent system to have explicit goals, we require processes that can generate or identify and select them dynamically. We shall refer to these processes as *goal formulation processes*. Where goals come from is often overlooked in intelligent system, which motivated us to address it in this survey.

Goal management: Third, the system should manage goals and select the ones that should be acted upon. An independent goal formulation process can lead to multiple goals. Therefore we require some form of management system that accepts goals produced by goal formulation processes, selects which goal(s) should be pursued (with reference to any ongoing goal-directed behavior), and triggers the appropriate plan generation mechanism to achieve the selected goal. If the goal formulation processes produce goals dynamically, asynchronously and in parallel, the management system must accept and manage new goals in this manner too. It should not block the operation of the goal formulation processes, as this would interfere with the system's ability to respond to new situations.

This set of requirements is consistent with those proposed by Hawes (2011). There is a fourth core requirement: the system should generate goal-directed behavior from a collection of goals and available resources. However, to simplify, we will ignore this requirement and assume that it is fulfilled by a planner with its execution system.

Our framework, GRAF (Table 1), is obtained by applying the five questions *What*, *Where*, *Why*, *When* and *How* to the three requirements of explicit goals, goal formulation and goal management.

Table 1. A tabular representation of GRAF.

Questions / Requirements	**What**	**Where**	**Why**	**When**	**How**
Explicit goals	Representation	Source			
Goal formulation			Rationale	Triggers	Methods
Goal management					Methods

What is a goal? This applies to the requirement of explicit goals and refers to the nature and representation of a goal. Explicit goals can be of two kinds. A *declarative* goal is a description of the state of the world which is sought and a *procedural* goal is a set of intended tasks to be solved. Consensus has it that most declarative goals are *attainment* goals. These are states an agent should achieve through plan execution. Declarative goals can also include *maintenance* and *prevention* goals, which refer to states to maintain over time or to prevent from occurring. Given our assumption that the required process which translates goals into behavior is a planning system, the nature of a goal and how it is explicitly represented in a system depends on that planner.

Where does a goal come from? This also applies to the requirement of explicit goals and refers to a goal's source. We identify three sources of goals: external, self, and hybrid. The goals can be supplied to the intelligent system by an *external* source in the environment (e.g., user or peer agents). Goals can also be *self-initiated* by the goal formulation process. While a majority of intelligent system designs assume the former, goal reasoning architectures focus on the latter. For the sake of completion, we also envision a *hybrid* situation where the goals can be both externally and internally initiated.

Why self-formulate a goal? This is applicable to the requirement of goal formulation. One reason to formulate goals is *rational anomaly response*: to better respond to developing situations that threaten an agent's interests. A second reason is *graceful degradation*: while the current goals may no longer be achievable, intelligent action may be achieved by degrading them (e.g., "submitting a full report" is predicted to fail given the time constraints, but "submitting a draft report" may be achievable). A third reason for goal formulation is better *future performance*: we want intelligent systems to avoid dead-ends with respect to the current goals, and also to avoid states that jeopardize goal achievement in the future. Furthermore, it may be desirable to take actions that increase the system's capabilities for more actions and more potential goals. A fourth reason for goal formulation is *societal norms*: as the scope of the agent's operation becomes broader and its lifespan longer, humans that interact with autonomous agents will have expectations about their behavior. Goals have to be accommodated to meet those expectations.

When is a goal formulated? This also applies to the requirement of goal formulation and refers to triggers for goal formulation. Typically, goal formulation is considered when an anomaly is detected and/or the system is self-motivated to explore its actions in the world.

How are goals formulated? This applies to the requirement of goal formulation and refers to methods for achieving the function of goal formulation.

How are goals managed? This also applies to the requirement of goal management and refers to methods for achieving the function of goal management.

In this survey, we primarily focus on the questions of *When* and *How*. That is, our emphasis is on triggers of goal formulation, methods for goal formulation, and methods for goal management.

3. Triggers for Goal Formulation

Typically, goal formulation can occur when an anomaly is detected and/or the system is self-motivated to explore its actions in the world. In most current implementations a goal is formulated when no active goal exists and the intelligent system is self-motivated to pursue additional goals, or an active goal exists but an anomaly is detected, and pursuing alternate goals is considered advantageous in light of the anomaly. Because a majority of existing approaches are anomaly-driven, we will focus on the latter. A non-exhaustive list of anomalies could include:

- An active plan fails (or is predicted to fail or perform suboptimally) and no contingency plan exists.
- An affordance is perceived (i.e., pursue a better goal that the agent was considering but hadn't been able to pursue).
- An opportunity is detected (i.e., pursue a better goal that the agent wasn't planning to pursue).
- An internal drive of a system requires attention (e.g., a battery's energy level is low and the system has an internal drive to maintain its energy level).

Anomaly triggered goal formulation requires a discussion about how anomalies are detected. Anomaly detection typically relies on various kinds of monitoring processes, including the following:

- **Plan monitoring**: One source of information for detecting anomalies comes from the plan itself. Changes in the environment may prevent the execution of a plan's future actions. In plan monitoring, the agent monitors the plan's execution by assessing whether its remaining actions' preconditions are satisfied in the current state or achievable as an effect of a preceding planned action. If not, a plan fails. Similarly, plans may also fail because an agent's actions do not achieve their intended effects. Action monitoring algorithms ensure that the last action was successfully executed (i.e., the effects of the action are true in the environment).

  In addition to monitoring the validity of plans during execution, research has identified methods for monitoring plan optimality during execution. Fitz & McIlraith (2007) define plan optimality and describe a state space planner that monitors the utility of the current plan with respect to alternatives using a variant of A* search. In this context, the agent should replan when it predicts that the plan will fail or execute sub-optimally.

Plan failure has been the subject of replanning and plan repair in traditional AI planning research from the beginning (Russell & Norvig, 2003). For example, Darmok implements action monitoring in an online case-based planner for a real-time strategy (RTS) game (Ontañon et al., 2010). If an action fails, Darmok extends its current plan with new actions to satisfy the failed action's goal. Also focusing on replanning, HOTRiDE employs action monitoring in simulated noncombatant evacuation operation planning (Ayan et al., 2007). When an action fails, HOTRiDE uses a dependency graph to determine which task decompositions are no longer valid and must be replanned.

When a plan fails or is predicted to fail (or be suboptimal), replanning systems try to generate new plans or repair existing plans using the original goal. In contrast, goal reasoning systems instead reason about their goals and try to formulate new goals. For example, ARTUE (Klenk et al., 2013) finds discrepancies (for discrete states) using a set difference operation between the expected and observed literals. For continuous states, the observed and expected value of each fluent is compared; a discrepancy is considered to occur whenever their values differ by more than 0.1% of the (absolute) observed value. When a discrepancy is detected, its anomaly response mechanism performs anomaly explanation and goal formulation.

- **Periodic monitoring**: Instead of focusing solely on the current plan and its execution, agents may monitor the entire environment to determine if new goals should be considered. In periodic monitoring, the agent considers the current state at set intervals. Periodic monitoring is frequently used in systems that perform real-time response. For example, Burkhard et al. (1998) illustrate how Belief-Desire-Intention (BDI) agents (Rao & Georgeff, 1995) monitor the environment for changes in their beliefs. Their RoboCup soccer agents receive new sensor information every 300ms. PROSOCS uses a sensing, revision, planning, and execution cycle to periodically monitor the environment (Mancarella et al., 2005). At the start of each cycle, new sensor information is received that can inform execution, plan revision, and future planning. A final example is the cognitive architecture ICARUS, which executes periodic monitoring during its recognize-act cycle (Langley & Choi, 2006).

- **Expectation monitoring**: Expectations are driven by experience from problem solving or interacting with an environment. Problem-solving experience can set expectations that can be monitored. A change in expectations can then trigger changes in behavior. For example, Veloso, Pollack and Cox (1998), in their rationale-based plan monitoring architecture, showed that plan rationales often include expectations that result in the adoption of the current plan at the expense of an alternative plan. Such expectations lead to (1) generating monitors that represent environmental features which affect plan rationale, (2) deliberating, whenever a monitor fires, about whether to respond to it, and (3) transforming plans as warranted by modifying goals. Expectation-driven goal-oriented behavior based on problem-solving experience is a hallmark of Schank's approach to intelligent systems (Schank 1982; Schank & Owens 1987), which is highly relevant to goal reasoning.

  Agents can also learn a model of how the environment changes through experience from interacting with their environment. Expectation monitoring uses this model to assess the nature and relevance of a discrepancy. In robotic navigation, Bouguerra, Karlsson, and Saffiotti (2008) used semantic knowledge to generate expectations concerning objects that may be encountered during plan execution. For example, when moving into a living room, the robot expects to see objects typical to that location (e.g., a TV, a sofa). From a cognitive science perspective, INTRO uses a rule-based model to generate expectations and detect discrepancies in a Wumpus World environment (Cox, 2007). Kurup et al. (2012) introduce a cognitive model of expectation-driven behavior in ACT-R. It generates future states called *expectations*, matches them to observed behavior, and reacts when a difference exists between them.

  Expectation monitoring can be implemented using anomaly recognition techniques. Typically, these approaches can be divided into three groups: (1) signature detection, which matches the current situation to known deviant patterns, (2) anomaly detection, which compares the current situation to baseline patterns, and (3) hybrid methods, which include both (Patcha & Park, 2007).

- **Domain-specific monitoring**: Monitoring for expectation failures is difficult in environments whose future states are difficult to predict. Therefore, some agents utilize domain-specific monitoring strategies, which periodically test values of specified state variables during plan execution. Many researchers use domain-specific monitoring to directly link unanticipated states to new goals. In a simulated rover domain, MADBot uses motivations to monitor specified values in the environment (e.g., when the battery's charge falls below 50%, a new goal is created to recharge it) (Coddington, 2006). M-ARTUE (Wilson, Molineaux, & Aha, 2013) similarly represents drives to direct goal formulation. While MADBot uses domain-specific drives, M-ARTUE does not represent motivations using domain knowledge, and is not limited to generating goals for achieving threshold values. Dora the Explorer (Hawes et al., 2011) encodes motivators that formulate goals related to exploring space and determining the function of rooms, similar to M-ARTUE's exploration motivator. However, Dora's functions are also domain-specific. Finally, Hawes's (2011) survey of motivation frameworks defines goal management and goal formulation in terms of goal generators or drives. It relates many systems in terms of these concepts, and proposes a design for future "motive management frameworks".
- **Object-based monitoring**: In domain-specific monitoring, the monitors specify particular state variables. Object-based monitoring also includes the set of objects in the environment. The detection of new objects may interrupt plans or cause the creation of new goals. Object-based monitoring systems specify which types of new objects to consider as discrepancies. Goldman (2009) describes an HTN planner with universally quantified goals that uses loops and other control structures to plan for sets of entities whose cardinality is unknown at planning time. Similarly, Cox and Veloso (1998) and Veloso, Pollack, and Cox (1998) also discuss and implement universally quantified goals where some objects (and hence goals) are not known. Dora generates a goal to explore each newly detected room (Hanheide et al., 2010). Open world quantified goals extend these approaches to include knowledge about how new objects may be detected (Talamadupula et al., 2010). For example, in an urban search and rescue task, plans must be generated to locate objects that are unknown prior to execution (i.e., the victims). In real-time games like GRUE (Gordon & Logan, 2004), a more typical approach for this kind of monitoring is by authoring game AI using a teleo-reactive program (TRP) (Benson & Nilsson, 1995). TRPs dictate which actions to take in specific world states (e.g., if the agent is running past a weapon it does not have, then it should pick up the weapon).

4. Methods for Goal Formulation

We identify six types of goal formulation methods based on the knowledge they use.

- **State-Based Goal Formulation**: The most straightforward method for generating goals is to pre-specify links between specific state variables and specific goals. Consider a helicopter's low-fuel indicator light. When it flashes, the agent pilot may generate a goal to refuel. The new goal depends solely on a single variable in the current state (i.e., the low-fuel indicator).

  These approaches are typically applied in fully observable environments. For example, game designers who have complete access to the environment can use behavior trees (Champandard, 2007) to control non-player characters; this is done in many modern video games. To increase reusability and make plans interruptible, Cutumisu & Szafron (2009) use multiple behavior trees to control characters interacting in a restaurant. Working with the internal state of the rover, AgentSpeak-MPL (Meneguzzi & Luck, 2007) uses motivations to formulate new goals when the value of particular state variables drops below individual thresholds. ICARUS (Choi 2010) uses a reactive goal management procedure to nominate and prioritize new top-level goals in which <condition, goal> pairs in long-term goal memory are considered for nomination at every reasoning step. This resembles rule-based goal-formulation, as used in ARTUE (Klenk et al. 2013). M-ARTUE (Wilson et al. 2013) includes a motivation subsystem that formulates goals based on the psychological notion of drives, which constitute a hierarchy of heuristic functions representing both external and internal needs. M-ARTUE differs from ARTUE only in the way goals are formulated; instead of using reactive rules, it uses domain independent heuristics to evaluate potential goals. This approach is similar in spirit to CLARION's goal formulation mechanism (Sun, 2009), where drives are represented sub-symbolically and they set the level of activation for explicit goals according to the world state. The primary difference between M-ARTUE and CLARION is that the representations of internal needs are domain independent and domain dependent, respectively.

- **Interactive Goal Formulation**: In realistic domains it is often infeasible to provide goal formulation knowledge for every situation. To address this, T-ARTUE (Powell, Molineaux, & Aha, 2011) and EISBot (Weber, Mateas, & Jhala, 2012) learn this knowledge from humans: T-ARTUE learns from criticism and answers to queries, while EISBot learns from human demonstrations. Each provides a domain-independent method for acquiring formulation knowledge, but neither system reasons about internal needs alongside external goals. Although based on the GDA model, GDA-C (Jaidee et al. 2013) differs substantially from ARTUE and M-ARTUE. GDA-C learns its goal selection function using Q-learning. While this increases autonomy, it employs a domain dependent reward function; indirectly, GDA-C's goal selection strategy is guided by a human.

- **Object-Based Goal Formulation**: While specifying a goal for each state provides an agent designer with considerable control over an agent's actions, these methods are inflexible and difficult to author. To promote reuse and flexibility, several systems rely on rules or schemas that specify how to formulate goals for a range of possible states. One important problem this solves is the generation of goals in response to the discovery of new objects in the environment that were unknown at planning time. Consider a robot on a search and rescue mission. Prior to plan execution, the number of rooms to search is unknown. Goal formulation allows the robot to formulate an initial plan to detect rooms, and then assert new goals to search the rooms as they are located.

  Recently, several researchers have proposed extensions to goal specifications to account for unknown objects. For example, goal generators produce goals when new objects are detected that satisfy a set of conditions (Hanheide et al., 2010). For example, when a new region is detected by a mobile robot, a goal will be generated to identify that region. In addition to generating goals based on newly detected objects, open-world quantified goals provide information about sensing actions for planning (Talamadupula et al., 2010). Each of these approaches extends the goal specification to specify the importance of the newly generated goal.

- **Belief-Based Goal Formulation**: In addition to the observed state, an agent may formulate goals using its beliefs about the current state. Representing knowledge about the environment that is not directly observed, beliefs are generally output by an inference process such as explanation or state elaboration. For example, on observing a lightning strike, an agent might infer a belief that a storm is approaching. This belief could lead to the formulation of a goal to seek shelter.

  Recent work has demonstrated the effectiveness of this approach in dynamic environments. After using explanation to update its beliefs, ARTUE uses rules to specify how to formulate goals based on the observed state and the agent's beliefs (Molineaux et al., 2010). An alternative method for generating beliefs is through state elaboration. Using forward inference rules over the observed state, ICARUS creates a set of beliefs, which are used by reactive goal management to nominate goals from long term memory for use in a simulated driving task (Choi, 2010).

- **Case-Based Goal Formulation**: Case-based goal formulation stores applicable goals in cases. During goal formulation, a case matching the cue is retrieved and the associated goal is reused in the current situation. For example, when a submarine disappears, an agent pilot might remember a previous situation in which searching for the submarine with a helicopter was a useful goal to pursue.

  Case-based goal formulation methods differ in their retrieval cues and types of goals generated. In EISBot (Weber, Mateas, & Jhala, 2010), the current state is used as a cue to retrieve a gameplay trace, which is a state sequence recorded from an expert's game play. EISBot selects a future state from the trace as the current goal. It performed well in StarCraft games against the built-in AI and human players. In another strategy game, CB-gda uses observed discrepancies as a retrieval cue to generate task goals (Muñoz-Avila et al., 2010). Each of these approaches requires minimal knowledge engineering as the retrieved cases may be automatically collected by observing human-provided traces of activities.

- **Explanation-Based Goal Formulation**: While the methods described above require knowledge engineering for each possible goal, an alternative approach focuses on explaining a discrepancy when generating a goal. When the observed discrepancy may prevent the agent from achieving its goals, the agent can generate a new goal by reasoning over its explanation. Consider a helicopter that is losing fuel. An agent pilot might explain this anomaly by inferring a leak in the fuel tank. Using this explanation, a goal could be generated to stop this leak.

Explanation-based methods use the explanation to generate goals. For example, INTRO (Cox, 2007) generates a goal by negating the antecedent of the explanation. In the Wumpus World domain, the discrepancy of the screaming wumpus would yield a goal to negate their hunger. In pervasive diagnosis, goals are generated to collect information based on the current diagnosis of faults in the device (Kuhn et al., 2008). The purpose is to generate plans to achieve production goals while refining its explanation for any faults. By focusing on the syntax of the explanation, these approaches can be easily applied to new domains.

Here we discuss four types of methods for explanation generation in response to an anomaly.

a. **Propositional Causal Models**: In such models, *p* causes *q* implies that *p* is always followed by *q*. A causal model is typically encoded as a set of rules, provided by a domain expert, which is used to infer the cause underlying a set of observations. This approach is exemplified by expert systems, such as the MYCIN medical diagnosis system (Shortliffe, 1976). Another deterministic approach uses truth-maintenance systems (Forbus & de Kleer, 1993), where facts are either assumptions provided to the system or consequences computed by a set of rules. For any consequence, it is possible to trace the rules and assumptions that support it.

   Intelligent agents have used deterministic causal models to improve their performance in problem-solving domains and simulated environments. For example, using explanation-based learning (DeJong, 1993), CASCADE applied overly-general rules to model human learning in physics problem solving (VanLehn et al., 1992). The goal reasoning agent ARTUE uses an abductive explanation (Josephson & Josephson, 1994) process to assume hidden facts that could cause a discrepancy (Molineaux et al., 2010). Using the environment model, ARTUE selects assumptions that, if true in the prior state, would predict the discrepancy *d* and the current state.

b. **Probabilistic Explanation Models**: Unlike deterministic models, probabilistic explanation models explicitly quantify uncertainty. In probabilistic models, *p* causes *q* implies that the occurrence of *q* increases the probability of *p*. Probabilistic explanation typically uses graphical models, such as Bayesian networks (Pearl, 2000), to determine the likely causes of individual propositions. These models rely on conditional independence between causes and the subjective probabilities can be learned by applying Bayes' rule with experience and a given prior probability. A probabilistic model of a ship explosion would include facts describing the likelihood of an explosion given a gas leak (or a fuel leak) as high, and the prior probability of a gas leak as higher than the prior probability of a torpedo. An agent would reason from this model that both a gas leak and a torpedo are possible explanations, with a gas leak being more likely.

   Probabilistic models have been adopted in many AI subfields. In planning under uncertainty, the environment is frequently modeled as a partially observable Markov decision process (Kaelbling, Littman, & Cassandra, 1998). A typical agent using this model will update an internal belief state after each action, which characterizes the probability of the agent being in each possible environment state. The update of this belief state is a form of explanation in which the observations are explained to result from a given state trajectory. From a goal reasoning perspective, pervasive diagnosis maintains a set of probabilities indicating the likelihood that each potential system fault has occurred based on prior observations (Kuhn et al., 2008).

c. **Qualitative Explanation Models**: This kind of model provides an alternative approach for describing uncertainty by allowing an agent to reason about changes to continuous quantities without using precise quantitative measurements. Quantity *q1* is qualitatively proportional to quantity *q2* if, all things being equal, an increase in *q1* causes an increase in *q2* (Forbus, 1984). A qualitative model may explain a ship explosion as the result of a decrease in the engine oil pressure that caused its temperature to rise above its flashpoint.

   Qualitative models are useful in domains where numerical models are unknown, inaccurate, or computationally expensive. For example, MAYOR (Fasciano, 1996) explains its expectation failures in managing a simulated city using a qualitative economic model (e.g., high crime decreases housing demand). Using a different qualitative economic model for cities, Hinrichs and Forbus (2007) use qualitative explanations to overcome local maxima in a worker placement task in the Freeciv turn-based strategy game.

d. **Example-specific Explanation Models**: Due to the difficulty of obtaining complete and correct models from domain experts (Watson, 1997), another approach is to rely on example-specific models, which are easier to elicit from experts. An expert may state that *p* causes *q* for a particular situation(s), and this knowledge may be used inductively to infer *p′* as a cause for *q′* in a new situation. For example, when faced with a new situation, case-based reasoning (Leake & McSherry, 2005) and analogical reasoning (Falkenhainer et al., 1989) approaches retrieve a similar example and reuse its example-specific explanation. Examples may be labeled with a cause, which can allow supervised learning approaches to infer causes for new instances (Mitchell, 1997). To explain a ship's explosion, an agent may recall another ship that was sunk by a submarine's torpedo and conclude that an enemy submarine is within range of the ship.

   The transfer of example-specific models has been used to improve the performance of AI systems. PHINEAS (Falkenhainer, 1988) creates analogies between qualitative behaviors to transfer explanatory models in physical domains. META-AQUA uses explanation patterns (Cox 2007), which are a type of case for explaining expectation violations. Muñoz-Avila and Aha (2004) define a taxonomy of explanation types pertinent to case-based planning for games.

5. Methods for Goal Management

In goal reasoning, agents may need to consider many goals. Given a set of pending goals, goal management selects which goal(s) should be pursued. Goal management can be a continuous ongoing process or triggered by certain events. For example, Veloso, Pollack and Cox (1998) discuss the use of rationale-based planning monitors as triggers for goal change, while Jones et al. (1999) represent goals as operators which are triggered at run-time by rules that match predefined states and sensor readings.

We identify seven types of plan-invariant methods (i.e., approaches that focus solely on pending goals) for goal management. They differ in how they store pending goals and how they select which goals to pursue. Shapiro et al. (2012) provide formal semantics for goal management by dropping or modifying intentions in the context of BDI agents, some of which are applicable to the methods discussed below.

- **Replacement**: Replacement remembers and plans for one goal at a time; if a new goal arises, it immediately replaces the existing goal. These approaches are useful when the set of goals is small, and the agent actively switches between them. For example, in Baltes's (2002) RoboCup soccer agent, the agent switches frequently between offense and defense based on the state of the field.
- **Stack (consider execution history)**: In lieu of strict replacement, an agent may use a stack to manage its goals. In this approach, the execution history is taken into account: a newly generated goal is accomplished first, after which the agent pursues the pending goals beginning with the goal that was being pursued when the most recent goal was generated. This is a common approach in cognitive architectures and other agents focused on long term execution. For example, both SOAR (Laird,

2008) and ACT-R (Anderson & Lebiere, 1998) agents have used this strategy to manage their goals in a wide range of domains. The same strategy is employed by the rover agents discussed previously (Coddington, 2006).

- **Rule-based (consider the state)**: In rule-based goal management systems, a set of rules is used to change the system's active goals. Each rule is a condition-action pair, where a condition is a statement about an event or a world state that, if true, results in an action to modify (e.g., add, drop, change) the current goals.

  Rule-based approaches have been used in reactive-planning agent architectures. While typical BDI agents (Rao & Georgeff, 1995) change their procedural goals as a result of observed events, CANPlan illustrates how observed events can trigger declarative goals that can be reasoned about using planning (Sardina & Padgham, 2010). Extending the semantics, the abstract agent language CAN specifies abstract goal states (pending, waiting, active, and suspended) for three different types of goals (achievement, task, and maintenance) and transitions among them (Harland et al., 2010).

- **Oversubscription planning (consider quantitative goals)**: Classical planning focuses on generating plans that achieve a conjunctive set of goals. If no such plan exists, then classical planning fails. Oversubscription planning (Smith, 2004) relaxes this all-or-nothing constraint, and instead focuses on generating plans that achieve the "best" subset of goals (i.e., the plan that gives the maximum trade-off between total achieved goal utility and total incurred action cost). While rule-based approaches do not include quantitative information in the goals themselves or how they are evaluated in a given state, oversubscription planning includes quantitative information in goals. This goal management strategy requires that each goal have an associated utility and each action have an estimated cost.

  While this greatly increases the computational complexity of finding an optimal plan, some heuristic approaches have been used for oversubscription planning. For example, heuristic Partial Satisfaction Planning approaches have been shown to generate plans of similar quality to optimal plans (van den Briel et al., 2004). Much of the research in this area has focused on describing the soft constraints that impact action costs and goal utilities. For example, *goal dependencies* (Do et al., 2007) involve constraints among goals (e.g., mutually exclusive goals), further complicating the goal selection process. While most oversubscription approaches do not consider changes to the agent's goals during execution, Han & Barber (2005) introduce a desire-space framework that accounts for goal dependencies. A desire-space is a Markov decision process (Sutton & Barto, 1998) in which each node is a set of achieved goals and the links between them are costs of a macro-operator that achieves the goals in the destination node. This enables the application of decision theory to determine which goals are worth the cost of achieving. Cushing, Benton, and Kambhampati (2008) describe an extension of oversubscription planning that includes replanning, which is cast as a process of reselecting goals. Each top-level goal is associated with rewards and penalties. Rewards are accrued when objectives are achieved and penalties otherwise. Newly arriving goals are modeled as rewards while existing plan commitments are modeled as penalties. The planner continually improves its current plan in an anytime fashion, while monitoring to see if any selected goal is still appropriate. Replanning occurs whenever a situation deviates significantly from the model, causing the selection of a new set of objectives.

- **Spreading activation (consider execution history and state)**: While the prior methods use only the time of the goal's formulation to determine the planner's goals, spreading activation methods determine the most relevant goals using the current context of the agent's working memory. In this approach, goals are associated with concepts in a semantic network. The concepts currently in working memory spread activation through the network to individual goals. The goal with the highest activation is selected for consideration by the agent. Motivated by psychological results which indicate that a goal stack insufficiently models human goal processing, some researchers have extended ACT-R's goal management system to select goals based on spreading activation in its declarative memory (Anderson & Douglass, 2001; Altmann & Trafton, 2002). Activation is spread between goals and cues based on associative links, which are formed when they enter working memory at the same time. This view of goal reasoning emphasizes the role of the environment to supply cues that activate the appropriate goals.

- **Priority queue (domain specific methods that incorporate execution history and state to prioritize goals)**: Priority queues generalize spreading activation to allow the ordering of goals along any preference metric (i.e., for each goal a number can be generated by some method using the current beliefs about the environment). The highest scoring goal is the one that should be pursued. Unlike the priority queue data structure, these approaches allow the priority of goals to change after being added to the queue. Therefore, each time an agent selects new goals, it must recompute the existing goals' priorities using its current beliefs about the environment.

  This approach has been used in research systems in robotics and game AI, some of which reason with learned priorities. For example, goal intensity allows a simulated rover agent to order its goals using the goals themselves and its beliefs about the environment (Meneguzzi & Luck, 2007). In robotics, the affective goal management method (Scheutz & Schermerhorn, 2009) maintains a recent history of previous successes and failures for each action type and uses these to estimate the expected utility for each goal. Instead of focusing solely on successes and failures, some systems incorporate appraisal theories (Roseman & Smith 2001). For example, the FearNot! framework selects goals related to the strongest emotions (Aylett, Dias, & Paiva, 2006), and SOAR 9 uses appraisals for intrinsically motivated reinforcement learning (Marinier, van Lent, & Jones, 2010). In game AI, GRUE (Gordon & Logan 2004) allows for concurrent goals to be pursued, but does so in a non-compensatory manner (i.e., goals with higher priorities receive preference for resources over all other goals). Similarly, the multi-queue approach to behavior trees (Cutumisu & Szafron, 2009) makes use of qualitative priorities between types of goals, and uses quantitative distinctions within each grouping to select the current goals. Young and Hawes's (2012) work on using evolutionary approaches to determine the priorities of high-level tasks in QUORUM also fits into this approach.

- **Goal transformation**: Goal transformation involves changing the current goals to enable plan generation (Cox & Veloso, 1998). Research on this topic has focused on defining the space of transfer formations and methods for applying them. For example, Cox & Veloso (1998) create a taxonomy of 13 goal transformations and demonstrate how they allow for graceful performance degradation in an air superiority planning task (e.g., in air combat planning, if insufficient resources are available to destroy a bridge, a new goal to damage the bridge can be generated). Goal Morph introduces costs and utilities to goal transformations in a web service composition application (Vukovic & Robinson, 2005). After constraining the space of applicable transformations using the context, Goal Morph applies the transformation that yields the goals with the highest utility.

6. Discussion

Goal formulation determines how an agent responds to an explained discrepancy. Many discrepancies do not require goal change. That is, the agent may continue executing the same plan, or it may generate a new plan for the same goals. While pure replanning approaches, such as FF-Replan, have been effective in many domains, they are susceptible to failures due to execution dead-ends (i.e., states from which the current goals cannot be achieved) (Yoon, Fern, & Givan, 2007). In addition to providing information about the environment, discrepancies may present threats to current or future plans, opportunities or obligations. One reason to formulate goals is to respond to developing situations that threaten the agent's interests, similar to the function of maintenance goals (Dastani, van Riemsdijk, & Meyer, 2006). There are other reasons for formulating goals: (1) graceful degradation, (2) improved future performance, and (3) societal norms. These other reasons have not been investigated sufficiently in goal reasoning research, which provides opportunities for future work.

With its focus on dynamic, uncertain, and open environments, goal reasoning seeks to increase autonomy through a knowledge intensive process. Therefore, goal formulation should not rely solely on the observed state, but also on the agent's beliefs about the environment, as in (Molineaux et al., 2010). In addition, it is difficult to specify all potential goals for an agent. Therefore, an important area of future research is to reduce the knowledge engineering burden by learning goal formulation methods, as in (Weber et al., 2010).

The need to consider competing goals is a primary motivation for goal reasoning. Simple replacement and stack approaches are well understood, but are too inflexible for more complex tasks. When planning failures occur, autonomous behavior requires a graceful degradation of performance, which may be achieved (at least partially) through existing oversubscription planning and goal transformation approaches. While oversubscription planning endows an agent with a rational method for selecting goals based on utility, it is insufficient when the set of goals is dependent on the agent's continuing observations of the environment (i.e., goals are subject to change at plan execution time). Approaches combining goal transformations with a definition of goal utility captured in a priority queue appear to be promising for handling larger classes of problems.

Future research should also investigate the interaction of goal reasoning components with traditional planning systems. Due to the separation of goal reasoning from planning, it should be possible to integrate a single goal reasoning method with multiple planners. Given that a state, a goal, and an environment model constitute a planning problem, it is worth exploring whether particular goal reasoning methods favor particular planners. In conducting this survey, we observed that the same or similar goal reasoning components may be used with the tasks of HTN planning (Molineaux et al., 2010) and the state-based goals used in many planning approaches (Hanheide et al., 2010). This suggests that goal reasoning is a distinct process worthy of independent investigation.

Evaluating goal reasoning systems is inherently difficult. AI researchers have produced many discussions on agent evaluation strategies (Kaminka & Burghart, 2007). In ablation experiments (e.g. Molineaux et al., 2010), a system's performance is evaluated through a series of trials during which components are removed to measure their contribution to the entire system. While there has been some research on discrepancy detection, explanation, goal formulation, and goal management, evaluating how each component performs within integrated intelligent systems will inform the design of future systems. Alternatively, Cassimatis, Bello, and Langley (2008) suggest comparing intelligent systems via metrics for capabilities, breadth, and parsimony. These metrics can provide evaluations based on a different view. Given the scope of the claims made about goal reasoning agents, a wide array of evaluation methodologies is needed to assess them.

7. Conclusion

Goal reasoning is motivated by four challenges to traditional planning approaches:

- *Nondeterministic partially observable environments*: An agent's observations of the current state are incomplete and the results of its actions are not deterministic. Furthermore, the environment may exhibit unbounded indeterminacy: it is not possible to fully enumerate the future states as a result of an agent's actions.
- *Dynamic environments*: The environment changes as a result of actions executed by the agent, events in the environment, or actions executed by other agents.
- *Incomplete knowledge*: In complex real-world domains, contingencies arise frequently but the knowledge of those contingencies may be limited. Furthermore, during execution, environment changes may present unidentifiable world states.
- *Knowledge engineering*: Capturing complete planning knowledge in complex real-world domains may require capturing wickedly large models for exogenous change, a prohibitively large number of contingencies, and probabilistic effects of actions. These can each present tremendous knowledge engineering challenges.

To enable intelligent action in these types of situations, we propose that agents should formulate and reason about their goals based on environmental changes. Goal reasoning is expected to provide two benefits to intelligent agents. First, goal reasoning should enable agents to better respond to unexpected circumstances. Second, goal reasoning should decrease the knowledge engineering burden in complex real-world domains for a given system by shifting the burden from capturing knowledge for exhaustive planning to that of coding models used by goals reasoning, which we conjecture to be an inherently simpler task. While there is some initial evidence supporting each claim (Handheide et al., 2010; Molineaux et al., 2010; Muñoz-Avila et al., 2010; Weber et al., 2010), further investigations are required.

As intelligent systems execute for longer periods without human intervention on a wide range of tasks, it becomes increasingly difficult to pre-specify all its possible goals and contingencies. Therefore, the current state-of-the-art relies on human operators to oversee an agent's execution on narrower tasks. But due to the proliferation of robotic and software agents in work, social, and residential environments, utilizing omnipresent human operators is not a viable option. Also, creating many systems for narrower tasks is inefficient and poses a usability challenge as people interact with each new system. Advances in goal reasoning should alleviate these bottlenecks to promote intelligent system development and deployment by increasing an agent's autonomy.

Acknowledgements

Thanks to OSD ASD (R&E) for sponsoring this research and to Michael Cox for his extensive recommendations. Swaroop Vattam performed this work while an NRC post-doctoral researcher located at NRL. The views and opinions contained in this paper are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of NRL or OSD.

References

Altmann, E. M., & Trafton, J. G. (2002). Memory for goals: An activation-based model. *Cognitive Science, 26,* 39–83.

Anderson, J.R., & Douglass, S. (2001). Tower of Hanoi: Evidence for the cost of goal retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 27,* 1331–1346.

Anderson, J.R., & Lebiere, C. (Eds.) (1998). *The atomic components of thought.* Hillsdale, NJ: Erlbaum.

Ayan, N.F., Kuter, U., Yaman F., & Goldman R. (2007). Hotride: Hierarchical ordered task replanning in dynamic environments. In F. Ingrand, & K. Rajan (Eds.) *Planning and Plan Execution for Real-World Systems – Principles and Practices for Planning in Execution: Papers from the ICAPS Workshop.* Providence, RI.

Aylett, R.S, Dias, J & Paiva, A. (2006). An affectively-driven planner for synthetic characters. *Proceedings of Sixteenth International Conference on Automated Planning and Scheduling* (pp. 2-10). Cumbria, UK: AAAI Press.

Baltes, J. (2002). Strategy selection, goal generation, and role assignment in a robotic soccer team. *Proceedings of the Seventh International Conference on Control, Automation, Robotics and Vision* (pp. 211-214). Singapore: IEEE Press.

Benson, S. & Nilsson, N. (1995). Reacting, planning and learning in an autonomous agent. In K. Furukawa, D. Michie, & S. Muggleton (Eds.) *Machine Intelligence 14*. Oxford, UK: Clarendon Press, Oxford.

Bouguerra, A. Karlsson, L., and Saffiotti, A. (2008). Active execution monitoring using planning and semantic knowledge. *Robotics and Autonomous Systems*. *56*(11), 942-954.

van den Briel, M., Sanchez Nigenda, R., Do, M.B., & Kambhampati, S. (2004). Effective approaches for partial satisfaction (over-subscription) planning. *Proceedings of the Nineteenth National Conference on Artificial Intelligence* (pp. 562-569). San Jose, CA: AAAI Press.

Burkhard, H. D., Hannebauer, M. & Wendler, J. (1998). Belief-desire-intention deliberation in artificial soccer. *AI Magazine*, *19*(3), 87-93.

Cassimatis, N., Bello, P., & Langley, P. (2008). Ability, breadth and parsimony in computational models of higher-order cognition. *Cognitive Science*, *32*(8), 1304-1322.

Champandard, A. (2007). Behavior trees for next-gen game AI. In *Proceedings of the Game Developers Conference*, Lyon. France.

Choi, D. (2010). *Coordinated execution and goal management in a reactive cognitive architecture*. Doctoral dissertation, Department of Aeronautics and Astronautics, Stanford University, Stanford, CA.

Coddington, A.M. (2006). Motivations for MADbot: A motivated and goal-driven robot. *Proceedings of the Twenty-Fifth Workshop of the UK Planning and Scheduling Special Interest Group* (pp. 39-46). Nottingham, UK: [http://www.cs.nott.ac.uk/~rxq/PlanSIG/PlanSIG06.htm].

Cox, M.T. (2007). Perpetual self-aware cognitive agents. *AI Magazine*, *28*(1), 32-45.

Cox, M.T., & Veloso, M.M. (1998). Goal transformations in continuous planning. In M. desJardins (Ed.), *Proceedings of the Fall Symposium on Distributed Continual Planning* (pp. 23-30). Menlo Park, CA: AAAI Press.

Cushing, W., Benton, J., & Kambhampati, S. (2008). *Replanning as a deliberative re-selection of objectives* (Technical Report). Computer Science and Engineering Department, Arizona State University, Tempe, AZ.

Cutumisu, M., & Szafron, D. (2009). An architecture for game behavior AI: Behavior multi-queues. *Proceedings of the Fifth AAAI Artificial Intelligence and Interactive Digital Entertainment Conference* (pp. 20-27). Stanford, CA: AAAI Press.

Dastani, M., van Riemsdijk, B., & Meyer, J.-J. (2006). Goal types in agent programming. *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 1285-1287). Hakodate, Japan: ACM Press.

DeJong, G. (1993). *Investigating explanation-based learning.* Norwell, MA: Kluwer Academic Publishers.

Do, M.B., Benton, J., van den Briel, M., & Kambhampati, S. (2007). Planning with goal utility dependencies. *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence* (pp. 1872-1878). Hyderabad, India: Professional Book Center.

Falkenhainer, B. (1988). *Learning from physical analogies* (Technical Report UIUCDCS-R-88-1479). Department of Computer Science, University of Illinois, Urbana-Champaign, IL.

Falkenhainer, B., Forbus, K.D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence, 41*(1), 1-63.

Fasciano, M.J. (1996). *Real-time case-based reasoning in a complex world* (Technical Report TR-96-05). Computer Science Department, the University of Chicago, Illinois.

Forbus, K. (1984). Qualitative process theory. *Artificial Intelligence*, *24,* 85-168.

Forbus, K. & de Kleer, J. (1993*). Building problem solvers.* Cambridge, MA: MIT Press.

Fritz, C., and McIlraith, S. A. (2007). Monitoring plan optimality during execution. *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling* (pp. 144-151). Providence, Rhode Island: ACM Press.

Goldman, R.P. (2009). Partial observability, quantification, and iteration for planning: Work in progress. In C. Fritz, S. McIlraith, S. Srivastava, & S. Zilberstein (Eds.) *Generalized Planning*: *Macros, Loops, Domain Control: Papers from the ICAPS Workshop*. Thessaloniki, Greece: [http://www.cs.berkeley.edu/~siddharth/genplan09/].

Gordon, E. & Logan, B. (2004). Game over: You have been beaten by a GRUE. In D. Fu & J. Orkin (Eds.) *Challenges in Game AI: Papers of the AAAI'04 Workshop* (Technical Report WS-04-04). San José, CA: AAAI Press.

Han, D. & Barber, K. (2005). Desire-space analysis and action selection for multiple dynamic goals. In *Computational Logic in Multi-Agent Systems* (pp. 249-264). Berlin: Springer.

Hanheide, M., Hawes, N., Wyatt, J., Göbelbecker, M., Brenner, M., Sjöö, K., Aydemir, A., Jensfelt, P., Zender, H., and Kruijff, G-J. (2010). A framework for goal generation and management. In D.W. Aha,

M. Klenk, H. Muñoz-Avila, A., Ram, & D. Shapiro (Eds.) *Goal-directed autonomy: Notes from the AAAI Workshop* (W4). Atlanta, GA: AAAI Press.

Harland, J., Thangarajah, J., Morley, D., and Yorke-Smith, N. (2010). Operational behaviour for executing, suspending, and aborting goals in BDI agent systems. In A. Omicini, S. Sardina, & W. Vasconcelos (Eds.) *Declarative Agent Languages and Technologies: Papers from the AAMAS Workshop.* Toronto, CA: [http://goanna.cs.rmit.edu.au/~ssardina/DALT2010].

Hawes, N. (2011). A survey of motivation frameworks for intelligent systems. *Artificial Intelligence, 175*(5-6), 1020-1036.

Hawes, N., Hanheide, M., Hargreaves, J., Page, B., Zender, H., & Jensfelt, P. (2011). Home alone: Autonomous extension and correction of spatial representations (pp. 3907-3914). In *Proceedings of the IEEE International Conference on Robotics and Automation*. Shanghai, China: IEEE Press.

Hinrichs, T.R., & Forbus, K.D. (2007). Analogical learning in a turn-based strategy game. *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence* (pp. 853-858). Hyderabad, India: Professional Book Center.

Jaidee, U., Munoz-Avila, H., & Aha, D.W. (2013). Case-based goal-driven coordination of multiple learning agents. *Proceedings of the Twenty-First International Conference on Case-Based Reasoning* (pp. 164-178). Saratoga Springs, NY: Springer.

Jones, R.M., Laird, J.E., Nielsen, P.E., Coulter, K.J., Kenny, P., & Koss, F.V. (1999). Automated intelligent pilots for combat flight simulation. *AI Magazine*, *20*(1), 27-41.

Josephson, J.R., & Josephson, S.G. (1994). *Abductive inference*. Cambridge, UK: Cambridge University Press.

Kaelbling, L.P., Littman, M.L. & Cassandra, A.R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence, 101*, 99-134.

Kaminka, G.A., & Burghart, C.R. (Eds.) (2007). *Evaluating architectures for intelligence: Papers from the AAAI Workshop* (Technical Report WS-07-04). San Mateo, CA: AAAI Press.

Klenk, M., Molineaux, M., & Aha, D.W. (2013). Goal-driven autonomy for responding to unexpected events in strategy simulations. *Computational Intelligence*, *29*(2), 187-206.

Kuhn, L., Price, B., de Kleer, J., Bo, M., & Zhou, R. (2008). Pervasive diagnosis: The integration of diagnostic goals into production plans. *Proceedings of the Twenty-Third Conference of the Association for the Advancement of Artificial Intelligence* (pp. 1306-1312). Chicago, IL: AAAI Press.

Kurup, U., Lebiere, C. Stentz, A. & Hebert, M. (2012). Using expectations to drive cognitive behavior. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. Ontario, Canada: AAAI Press.

Laird, J.E. (2008). Extending the Soar cognitive architecture. *Proceedings of the First Artificial General Intelligence Conference* (pp. 224-235). Memphis, TN: IOS Press.

Langley, P., & Choi, D. (2006). A unified cognitive architecture for physical agents. *Proceedings of the Twenty-First National Conference on Artificial Intelligence*. Boston, MA: AAAI Press.

Leake, D. (1991). Goal-based explanation evaluation. *Cognitive Science, 15*, 509-545.

Leake, D. B., & Ram, A. (1995). Learning, goals, and learning goals: a perspective on goal-driven learning. *Artificial Intelligence Review, 9*(6), 387-422.

Leake, D. & McSherry, D. (2005). Introduction to the special issue on explanation in case-based reasoning. *Artificial Intelligence Review, 24*(2), 103-108.

Mancarella, P., Sadri, F., Terreni, G., & Toni, F. (2005). Planning partially for situated agents. In *Computational Logic in Multi-Agent Systems* (pp. 230-248). Berlin: Springer.

Marinier, B., van Lent, M., and Jones, R. (2010). Applying appraisal theories to goal directed autonomy. In D.W. Aha, M. Klenk, H. Muñoz-Avila, A., Ram, & D. Shapiro (Eds.) *Goal-directed autonomy: Notes from the AAAI Workshop* (W4). Atlanta, GA: AAAI Press.

Meneguzzi, F.R., & Luck, M. (2007). Motivations as an abstraction of meta-level reasoning. *Proceedings of the Fifth International Central and Eastern European Conference on Multi-agent Systems* (pp. 204-214). Leipzig, Germany: Springer.

Mitchell, T. (1997). *Machine learning*. Columbus, Ohio: McGraw-Hill.

Molineaux, M., Klenk, M., & Aha, D.W. (2010). Goal-driven autonomy in a Navy strategy simulation. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*. Atlanta, GA: AAAI Press.

Muñoz-Avila, H., & Aha, D.W. (2004). On the role of explanation for hierarchical case-based planning in real-time strategy games. In D. McSherry & P. Cunningham (Eds.), *Explanation in Case-Based Reasoning: Papers from the ECCBR Workshop* (Technical Report 142-04). Madrid, Spain: Universidad Complutense de Madrid, Departamento de Sistemas Informáticos y Programación.

Muñoz-Avila, H., Jaidee, U., Aha, D.W., & Carter, E. (2010). Goal directed autonomy with case-based reasoning. *Proceedings of the Eighteenth International Conference on Case-Based Reasoning* (pp. 228-241). Alessandria, Italy: Springer.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Norman, T.J., & Long, D. (1996). Alarms: An implementation of motivated agency. In *Intelligent Agents II: Agent Theories, Architectures, and Languages* (pp. 219-234). Berlin: Springer.

Ontañon, S., Mishra, K., Sugandh, N., & Ram, A. (2010). On-line case-based planning. *Computational Intelligence*, *26*(1), 84-119.

Patcha, A., & Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, *51*, 3448-3470.

Pearl, J. (2000). *Causality: Models, reasoning and inference*. Cambridge, UK: Cambridge University Press.

Powell, J., Molineaux, M., & Aha, D.W. (2011). Active and interactive learning of goal selection knowledge. In *Proceedings of the Twenty-Fourth Florida Artificial Intelligence Research Society Conference*. West Palm Beach, FL: AAAI Press.

Rao, A., & Georgeff, M. (1995). BDI agents: From theory to practice. *Proceedings of the First International Conference on Multi-agent Systems* (pp. 312-319). Menlo Park, CA: AAAI Press.

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.

Russell, S., & Norvig, P. (2003). *Artificial intelligence: A modern approach* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Sardina, S., & Padgham, L. (2010). A BDI agent programming language with failure handling, declarative goals, and planning. *Autonomous Agents and Multi-Agent Systems*, *23*(1), 18-70.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Schank, R. C. (1982). *Dynamic memory: A theory of reminding and learning in computers and people*. Cambridge, MA: Cambridge University Press.

Schank, R. C., & Owens, C. C. (1987). Understanding by explaining expectation failures. In R. G. Reilly (Ed.), *Communication Failure in Dialogue and Discourse*. New York: Elsevier Science.

Scheutz, M. & Schermerhorn, P. (2009). Affective goal and task selection for social robots. In J. Vallverdú & D. Casacuberta (Eds.) *The Handbook of Research on Synthetic Emotions and Sociable Robotics*. Hershey, PA: IGI Publishing.

Shapiro, S., Sardina, S., Thangarajah, J., Cavedon, L., & Padgham, L. (2012). Revising conflicting intention sets in BDI agents. *Proceedings of the Eleventh International Conference on Autonomous*

Agents and Multiagent Systems (pp. 1081-1088). Valencia, Spain: International Foundation for Autonomous Agents and Multiagent Systems.

Shortliffe, E.H. (1976). *Computer-based medical consultations: MYCIN*. New York: Elsevier/North Holland.

Smith, D.E. (2004). Choosing objectives in over-subscription planning, *Proceedings of Fourteenth International Conference on Automated Planning and Scheduling* (pp. 393 – 401). Whistler, British Columbia, Canada: AAAI Press.

Sun, R. (2009). Motivational representations within a computational cognitive architecture. *Cognitive Computation, 1*(1), 91-103.

Sutton, R.S., & Barto, A.G. (1998). *Reinforcement learning: An introduction.* Cambridge, MA: MIT Press.

Talamadupula, K., Benton, J., Kambhampati, S., Schermerhorn, P., & Scheutz, M. (2010). Planning for human-robot teaming in open worlds. *ACM Transactions on Intelligent Systems and Technology*, *1*(2), Article 14.

VanLehn, K., Jones, R. M., and Chi, M. T. H. (1992). A model of the self-explanation effect. *Journal of the Learning Sciences*, *2*(1), 1-59.

Veloso, M. M., Pollack, M. E., & Cox, M. T. (1998). Rationale-based monitoring for continuous planning in dynamic environments. In R. Simmons, M. Veloso, & S. Smith (Eds.), *Proceedings of the Fourth International Conference on Artificial Intelligence Planning Systems* (pp. 171-179). Menlo Park, CA: AAAI Press.

Vukovic, M., & Robinson, P. (2005). GoalMorph: Partial goal satisfaction for flexible service composition. *International Journal of Web Services Practices, 1*(1-2), 40-56.

Watson, I. (1997). *Applying case-based reasoning: techniques for enterprise systems*. San Francisco, CA: Morgan Kaufmann.

Weber, B., Mateas, M., & Jhala, A. (2010). Applying goal-driven autonomy to StarCraft, In *Proceedings of Sixth Artificial Intelligence and Interactive Digital Entertainment*. Stanford, CA: AAAI Press.

Weber, B., Mateas, M., & Jhala, A. (2012). Learning from demonstration for goal-driven autonomy. In P*roceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. Toronto, Canada: AAAI Press.

Wilson, M., Molineaux, M., & Aha, D.W. (2013). Domain-independent heuristics for goal formulation. In *Proceedings of the Twenty-Sixth Florida Artificial Intelligence Research Society Conference*. St. Pete Beach, FL: AAAI Press.

Yoon, S., Fern, A., and Givan, B. (2007). FF-replan: A baseline for probabilistic planning. *Proceedings of Seventeenth International Conference on Automated Planning and Scheduling* (pp. 352-359). Providence, Rhode Island: ACM Press.

Young, J. and Hawes, N. (2012) Evolutionary Learning of Goal Priorities in a Real-Time Strategy Game. In *Proceedings of the Eighth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. Stanford, CA: AAAI Press.