

Goal-Driven Autonomy in Planning and Acting

Matthew Klenk

Navy Center for Applied Research in Artificial Intelligence

Naval Research Laboratory (Code 5514)

Washington, DC, 20375

matthew.klenk.ctr@nrl.navy.mil

Abstract

To operate autonomously in complex environments, agents must perform actions, sense the environment, and respond to new situations. Traditional approaches face difficulties with incomplete environment models, goal specification, and engineering domain specific control knowledge. We believe goal reasoning will address these challenges enabling agents to better respond to unexpected circumstances and reduce the knowledge engineering burden. We present a conceptual model of *goal-driven autonomy* (GDA) integrating four reasoning tasks: discrepancy detection, explanation, goal formulation, and goal management. Additional work is required to understand the implications of goal reasoning, and GDA in particular, for improving the robustness of autonomous agents.

Introduction

To operate autonomously in complex environments, agents must perform actions, sense the environment, and respond to new situations. Traditional planning approaches for selecting actions assume complete knowledge of user-specified goals. Intelligent agent approaches employ sub-goaling within sensing and action cycles. These approaches confront knowledge engineering challenges with respect to modeling the environment, specifying the goals, and engineering the domain specific search control knowledge. To overcome these difficulties, we present a conceptual model of *goal-driven autonomy* (GDA). GDA performs goal reasoning in four tasks: discrepancy detection, explanation, goal formulation, and goal management. If successful, goal reasoning should enable agents to better respond to unexpected circumstances and decrease the knowledge engineering burden of system designers.

To illustrate the importance of goal reasoning, consider a fishing craft in the Gulf of Mexico. While carrying out a plan to achieve the goal of catching fish at a particular location, it receives reports that an offshore oil rig has suffered an explosion. At this point, the ship will change its goal to rescuing 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.

Aspirations of Goal Reasoning

In this section, we argue for research on goal reasoning approaches, such as GDA, focusing knowledge engineering challenges of real world applications. Addressing these challenges through goal reasoning should result in more robust agents, and, at the end of this section, we discuss a few ways to measure research progress.

Why Goal Reasoning?

Current approaches for constructing intelligent agents require significant knowledge engineering to model the environment, goals, and control knowledge. While models are necessarily incomplete with respect to the phenomena in question, environments most suitable for goal reasoning are dynamic, unbounded and open with respect to the introduction of new objects. In dynamic environments, the state changes not only from the actions of the agent, but also as a result of actions taken by other agents and events occurring in the environment. Unbounded indeterminacy (Russell & Norvig 2003) characterizes environments where not only are future states unpredictable, but the complete set is either too large to enumerate or unknown to the agent. Finally, the complete set of agents or world objects is unknown to the agent. Therefore, the agent must reason about unknown, unobserved, and new objects throughout planning and execution.

In addition to difficulties in modeling the environment, it is often difficult to develop a complete model of the goals or possible goals. This partially results from the environment. For example, some of the unknown objects may be goal relevant. Furthermore, unanticipated changes in the environment may require reevaluating the utilities given to the goals. This problem becomes increasingly difficult when considering the effects of goal dependencies (Do *et al.* 2007) on utility.

The third motivation for goal reasoning is decreasing the amount of knowledge engineering required for action selection methods. The knowledge engineering for planning has received considerable attention in recent years, including a competition at ICAPS-2009 (Bertoli *et al.* 2009). For agent approaches using HTN planning (Erol *et al.* 1994) and subgoaling (e.g., Laird 2008), the domain

specific control knowledge is encoded in their methods. This complicates the semantics of task decomposition to include reasoning about which task to perform in the current environment.

Hypotheses about the Future

From these motivations, we consider two hypotheses concerning goal reasoning.

Hypothesis 1: Goal reasoning enables agents to better respond to unexpected circumstances.

Gathering evidence concerning this hypothesis has two difficulties. (1) It is difficult to provide performance metrics and scenarios that do not unfairly handicap ablated versions of the system. It is highly unlikely that a system that does not generate a specific goal will be able to achieve it. (2) When goal reasoning is ablated from a system, how are the other components of the system affected. Specifically, it is difficult to measure exactly how much domain knowledge has been removed from the system.

Initial investigations of goal reasoning have provide support of this hypothesis by ablating the goal reasoning aspects of their systems (ARTUE, Molinueax *et al.* 2010; DORA, Hanheide *et al.* 2010; GDA-HTNbots Munoz-Avila *et al.* 2010). Another approach would be to look at relative performance as domain knowledge is removed from the system. This would result in more unexpected events and consequently, the goal reasoning system’s performance should degrade less than a comparable system without goal reasoning.

Hypothesis 2: Goal reasoning decreases the knowledge engineering burden for a given system.

The incorporation of goal reasoning should reduce the complexity of an agent’s subgoaling or planning methods. Measuring the knowledge engineering burden of a system is difficult. Knowledge engineering expertise often confounds typical metrics, such as the domain encoding time. An interesting proxy for the knowledge engineering effort required is the ability for automatic methods to learn the necessary domain knowledge. While none of the initial goal reasoning investigations has addressed learning, this seems like a promising future direction. Perhaps the most impressive evidence for this hypothesis may come from constructing systems capable of robust action in increasingly difficult environments.

Goal-Driven Autonomy

Cox’s (2007) INTRO system provides the inspiration for several concepts in GDA, with its focus on integrated planning, execution, and goal reasoning. We extend these ideas and integrate them with Nau’s (2007) *online planning* framework.

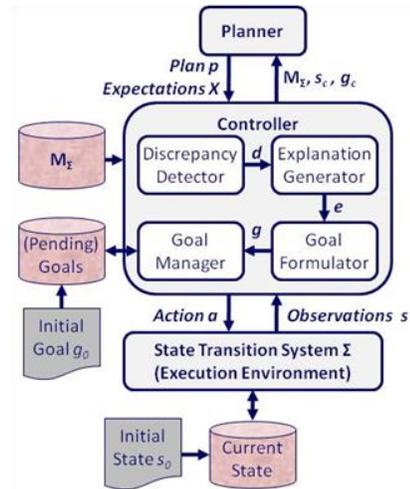


Figure 1: Conceptual model of Goal-Driven Autonomy

As shown in Figure 1, GDA is a conceptual model of online planning in autonomous agents. The GDA model primarily expands and details the scope of the *Controller*, which interacts with a *Planner* and a *State Transition System* Σ (an execution environment).

System Σ is a tuple (S, A, E, γ) with states S , actions A , exogenous events E , and state transition function $\gamma: S \times (A \cup E) \rightarrow 2^S$, which describes how an action’s execution or an event’s occurrence transforms the environment from one state to another. In real environments, the agent has only partial knowledge of the state, set of events, and state transition function.

The Planner receives as input a planning problem (M_Σ, s_c, g_c) , where M_Σ is a model of Σ , s_c is the current state, and g_c is a goal that can be satisfied by some set of states $S_g \subseteq S$. The Planner outputs a plan p_c , which is a sequence of actions $A_c = [a_c, \dots, a_{c+n}]$ that result in satisfying g_c . In the GDA model, the Planner generates a corresponding sequence of expectations $X_c = [x_c, \dots, x_{c+n}]$, where each $x_i \in X_c$ is a set of state constraints corresponding to the sequence of states $[s_{c+1}, \dots, s_{c+n+1}]$ expected to occur when executing A_c beginning in s_c using M_Σ .

The Controller sends the actions in the plan to Σ and processes the resulting observations. In the GDA model, it takes as input initial state s_0 , initial goal g_0 , and M_Σ , which are all sent to the Planner to generate a plan p_0 with associated expectations X_0 . When executing p_0 , the Controller performs the following four knowledge-intensive tasks, which uniquely distinguish the GDA model: *discrepancy detection*, *explanation generation*, *goal formulation*, and *goal management*.

Discrepancy Detection

GDA must first detect unexpected changes to the environment before deciding how to respond to them. This task compares the current observations s_{c+1} obtained from executing action a_c in state s_c with the expectation x_c generated by the planner. If a non-empty set of

discrepancies D (i.e., mismatches between the expectation x_c the observed state s_{c+1}) are found, explanation generation is performed to explain them.

Discrepancy detection approaches differ in terms of the expectations x_c considered. *Execution monitoring* approaches focus solely on the current plan p being executed. These include expectations that each of p 's actions achieve their intended effects and that the p 's remaining preconditions remain true (Russell & Norvig 2003). *World state monitoring* approaches include expectations about either the entire state, or subsets not directly related to the current plan. World state monitoring approaches allow an agent to take advantage of opportunities as well as avoid problems by recognizing situations in which goal change is useful.

Explanation Generation

The cause for a detected discrepancy must be considered so that it can be addressed. Given the previous and observed states, s_c and s_{c+1} , and a set of discrepancies D , this task hypothesizes one or more explanations e of their cause. An explanation may concern the state s_{c+1} , the expectations x_c , or agent's environment model M_S .

While AI research includes many methods for generating explanations including rule-based expert systems (Shortliffe 1976), inductive reasoning programs (Muggleton 1991), probabilistic graphical models (Pearl 2000), and model-based diagnosis (Falkenhainer & Forbus 1991), there have been few approaches using explanation to improve an agent's performance. Notable exceptions include Explanation Based Learning (DeJong 1993) where explanation is used to infer problem-solving control knowledge from an example. In the context of goal-driven autonomy, explanation is an essential component for providing an agent with information regarding the current situation.

Goal Formulation

Resolving the discrepancy may require generating new goals for consideration. Goal formulation generates goal(s) g in response to D , given explanation(s) e and the believed current state s_{c+1} . Not all discrepancies require goal formulation; in fact many discrepancies may be ignored, or their explanations may involve assumptions not relevant to the current plan or potential goals of the agent.

Goal formulation methods differ in terms of *how* they generate their goals and *why* they generate them. Goal generation methods may rely on links between individual states and pre-specified goals (Dill & Papp 2005), schemas instantiating new goals based on the current state (Talamadupula *et al.* 2009), and on the explanation for the discrepancy itself (Cox 2007). More relevant to GDA is the purpose of the generated goal. In particular, goal formulation approaches should enable the agent to take advantage of opportunities as well as avoid potential obstacles due to unanticipated changes in the environment.

Goal Management

The formulation of a new goal may warrant its immediate focus and/or the removal of some existing goals. Given a set of pending goals G_P , one or more of which is the focus of the current plan, and new goal(s) g , goal management may update G_P by adding g and/or deleting/modifying other pending goals and will select the next goal(s) $g' \in G_P$ to be given to the Planner. It is possible that the goal will not change, $g'=g_c$, or the generated goal will be selected, $g'=g$.

AI researchers have considered a wide range of goal management approaches. Simple approaches include replacing the previous goal with the formulated goal (Baltes 2002) and maintaining a goal stack in which the new goal is addressed first (Coddington 2006). Knowledge intensive approaches consider the utility of the new goals (Meneguzzi & Luck 2007) as well as the expected costs to achieve them (Smith 2004). Goal management may include additional processes other than goal selection. For example, *goal transformations* may be useful for maintaining performance when no plan exists for the original goals (Cox & Veloso 1998).

Relation to Other Approaches

Goal reasoning is intended to assist in action selection for environments that from the agent's perspective are non-deterministic. There have been other approaches within the planning and agents community to address this. *Contingency planning* (Dearden *et al.* 2003) agents generate conditional plans that are executed only when an action does not achieve its intended effects. Plans may also fail due to unanticipated changes in the environment. To avoid reasoning about unanticipated changes, *incremental planners* plan for a fixed time horizon, execute the plan, and then generate a new plan from the current state. This process continues until a goal state is reached. For example, CPEF (Myers 1999) generates plans to achieve air superiority in military combat by replanning periodically and when unexpected events occur during execution (e.g., a plane is shot down).

In addition to GDA, there have been a number of approaches for goal reasoning. The importance of maintaining separate goal reasoning was first identified and demonstrated within the Prodigy architecture as a response to failures in plan generation (Cox & Veloso 1998). From a planning perspective, efforts to allow for greater expressivity in goals include Talamadupula *et al.*'s (2009) *open world quantified goals*, which describe goals for unknown objects and Goldman's (2009) universally quantified goals that allows planning for sets of entities whose cardinality is unknown at planning time. These approaches place all goal reasoning in the planner itself. An alternative is to define structures for goal reasoning during execution. For example, Coddington and Luck (2004) bestowed agents with *motivations*, which generate goals in response to specific observed states. For example,

if a rover's battery charge falls below 50%, then a goal to recharge the battery will be generated (Meneguzzi & Luck 2007).

Intelligent agents typically employ reasoning cycles to sense the environment and perform actions. Choi (2010) has extended the ICARUS architecture (Langley & Choi 2006) so that it can create goals using constraint-like goal descriptions. SOAR agents (Laird 2008) employ subgoaling and chunking to acquire new skills for goals that are not directly attainable. Salvucci and Taatgen's (2008) theory of *threaded cognition* demonstrates how ACT-R (Anderson 2007) can perform concurrent multitasking by introducing new goals based on the current beliefs. While these agents all perform goal reasoning, their conceptual frameworks are not explicitly focused on this process.

Conclusions

Applying traditional approaches for action selection relies on complete knowledge of user-specified goals. In particular, these approaches involve three knowledge engineering challenges: modeling the environment, encoding the complete set of potential goals, and specifying domain specific control knowledge. In overcoming these obstacles, goal reasoning should enable agents to respond competently to unanticipated changes in the environment and reduce the knowledge engineering complexity of the agent itself. We present goal-driven autonomy as a conceptual model of goal reasoning integrating discrepancy detection, explanation, goal formulation and goal management. While initial experiments have demonstrated the promise of goal reasoning (Molineaux *et al.*, 2010), additional work remains for understanding how different approaches may be applied for different applications. Advances in goal reasoning should increase the level of autonomy exhibited by intelligent agents in increasingly complex tasks and environments including autonomous vehicles and home assistance robots.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford University 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.

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 directed 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/coddington.pdf>]

Coddington, A., & Luck, M. (2004). A motivation-based planning and execution framework. *International Journal on Artificial Intelligence Tools*, **13**(1), 5-25.

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.

Dearden R., Meuleau N., Ramakrishnan S., Smith, D., & Washington, R. (2003). Incremental contingency planning. In M. Pistore, H. Geffner, & D. Smith (Eds.) *Planning under Uncertainty and Incomplete Information: Papers from the ICAPS Workshop*. Trento, Italy.

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

Dill, K., & Papp, D. (2005). A goal-based architecture for opposing player AI. In *Proceedings of the First Artificial Intelligence for Interactive Digital Entertainment Conference* (pp. 33-38). Marina del Rey, CA: AAAI Press.

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.

Erol, K., Nau, D., & Hendler, J. (1994). HTN planning: Complexity and expressivity. *Proceedings of the Twelfth National Conference on Artificial Intelligence* (pp. 123-1128). Seattle, WA: AAAI Press.

Falkenhainer, B., & Forbus, K. (1991). Compositional modeling: Finding the right model for the job. *Artificial Intelligence*, **51** (1-3), 95-143.

Hanheide, M., Hawes, N., Wyatt, J., Gobelbecker, M., Brenner, M., Sjo, K., Aydemir, A., Jenselt, P., Zender, H. and Kruijff, G. (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: Papers from the AAAI Workshop*. Unpublished working notes.

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. In *Proceedings of the Twenty-First AAAI Conference on Artificial Intelligence*. Boston, MA: AAAI Press.

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

Muggleton, S.H. (1991). Inductive Logic Programming. *New Generation Computing*, **8**(4), 295-318.

Munoz-Avila, H., Aha, D.W., Jaidee, U., & Carter, E. (2010). Goal-driven autonomy with case-based reasoning. To appear in *Proceedings of the Eighteenth International Conference on Case-Based Reasoning*. Alessandria, Italy: Springer.

Myers, K.L. (1999). CPEF: A continuous planning and execution framework. *AI Magazine*, **20**(4), 63-69.

Nau, D.S. (2007). Current trends in automated planning. *AI Magazine*, **28**(4), 43-58.

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

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

Salvucci, D.D., & Taatgen, N. A. (2008). Threaded cognition: An integrated theory of concurrent multitasking. *Psychological Review*, **115**, 101-130.

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 the Fourteenth International Conference on Automated Planning and Scheduling* (pp. 393-401). Whistler (BC), Canada: AAAI Press.

Talamadupula, K., Benton, J., Schermerhorn, P., Kambhampati, S., & Scheutz, M. (2009). Integrating a closed world planner with an open world robot: A case study. In M. Likhachev, B. Marthi, C. McGann, & D.E. Smith (Eds.) *Bridging the gap between action and motion planning: Papers from the ICAPS Workshop*. Thessaloniki, Greece.