Human-Agent Teaming as a Common Problem for Goal Reasoning

Matthew Molineaux,[1] **Michael W. Floyd,**[1] **Dustin Dannenhauer,**[2] **David W. Aha**[3]
[1]Knexus Research Corporation; Springfield, VA | {first.last}@knexusresearch.com
[2]NRC Postdoctoral Fellow; NRL; Navy Center for Applied Research in AI; Washington, DC | dustin.dannenhauer.ctr@nrl.navy.mil
[3]Naval Research Laboratory; Navy Center for Applied Research in AI; Washington, DC | david.aha@nrl.navy.mil

Abstract

Human-agent teaming is a difficult yet relevant problem domain to which many goal reasoning systems are well suited, due to their ability to accept outside direction and (relatively) human-understandable internal state. We propose a formal model, and multiple variations on a multi-agent problem, to clarify and unify research in goal reasoning. We describe examples of these concepts, and propose standard evaluation methods for goal reasoning agents that act as a member of a team or on behalf of a supervisor.

1 Introduction

An important focus of research on intelligent agents is to achieve goals quickly and reliably. In recent years, goal reasoning researchers have considered the issue of *goal change*, a process by which an agent can shift the overall focus of its activities. This change can be prompted by a nameless outside goal source and/or an internal motivation model. In this work, we advocate modeling the other agents whose goals an agent attempts to achieve. With this model change, it becomes clear that goal reasoning agents are particularly well-suited to being team players. We define a human-agent teaming model and problem, and discuss how future goal reasoning research can leverage it.

Research on goal reasoning has investigated multiple framework abstractions for algorithms and agent architectures (e.g., Goal-Driven Autonomy (Molineaux, Klenk, and Aha 2010) and the Goal Lifecycle (Roberts et al. 2014)), but has not focused on common problems. Areas such as reinforcement learning and automated planning have benefited greatly from such a focus, receiving additional attention from competitions and comparing results via easy-to-use benchmarks. While one problem may not suffice to compare all goal reasoning agents, a small number of common problems could facilitate comparative publications, and thereby focus goal reasoning research. This paper focuses on elaborating this position, and a candidate formal framework for describing classes of problems; we expect that future work will specify concrete representations and initial problems.

In Section 2, we provide a formal description of a general *human-agent teaming* problem, along with several important variations that are commonly encountered in goal reasoning research. We then discuss some examples of the concepts described in Section 3, and discuss useful metrics for comparison in Section 4. Finally, in Section 5 we conclude.

2 Models of Goal Reasoning for Human-Agent Teaming

In recent work, goal reasoning systems have explicitly reasoned over the presence of other agents and their goals. For example, goal reasoning agents may be aware that their opponent in a real-time strategy game is attempting to defeat them (Weber, Mateas, and Jhala 2010; Jaidee, Muñoz-Avila, and Aha 2013; Dannenhauer and Muñoz-Avila 2015), that other agents may attack them (Bonnano et al. 2016), or that other agents may impede them (Cox 2013). Other work has described explicit exchange of goals and other information between agents and humans for the purpose of general collaborative tasks (Geib et al. 2016), control of unmanned vehicles (Richards and Stedmon 2017), and autonomous community formation (Golpayegani and Clarke 2016). The framework presented here is designed to facilitate communication and comparison of agents that work together in these ways. Concepts described here help with the modeling of the goals, plans, and motivations of other agents, especially those that reason over goals themselves. In the spirit of the successful reinforcement learning problem (Sutton and Barto 1998), we describe a simple set of functions and informational items intended to be general enough to be easily applied and used by all agents that solve these problems. In order to keep this framework generic and approachable, we avoid committing to representations and functions that many agents may not be able to provide.

In our model (Figure 1), a team is situated in an environment. This team can comprise goal reasoning agents, human teammates, and other software agents. At each time t ($t \in T$, the set of discrete time points at which communications occur), each *teammate* observes the environment. The environment's state is given by s_t ($s_t \in S$, the set of all environment states), and teammate m ($m \in M$, the set of teammates) receives an observation o_t^m ($o_t^m \in O$, the set of all observations). The environment creates individualized observations for each agent; we model the observation generation process as a function $obs^m : S \to O$. Teammates can perform

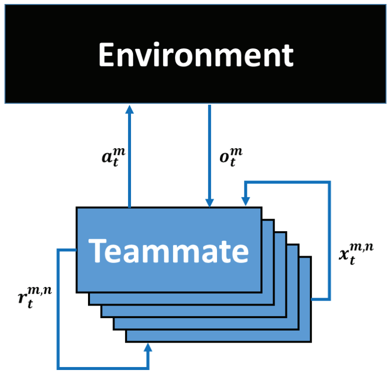

Figure 1: Human-Agent Teaming Problem Model

an action at each time t, denoted a_t^m ($a_t^m \in A$, the set of all actions). Changes in the environment are dependent on these actions as well as the prior state, which we model as the transition function $\lambda : S \times A^{|M|} \to S$. This generic representation allows for description of a wide variety of environments, including those with heterogeneous observability, exogenous events, and role-based actions; however, it does not permit continuous time.

Acting as teammates imposes some extra requirements on an agent. Work in human factors (Klein et al. 2004) has recognized four distinct requirements for acting as a member of team. Loosely summarized, they are: (1) agree on common goals; (2) direct and take direction from other teammates; (3) predict the behavior of other teammates and act in a way they can predict; and (4) maintain a common understanding of the shared environment. To support these requirements, in our framework teammates communicate via requests and explanations. A teammate m can make a *request* $r_t^{m,n}$ of another teammate $n \in M$ ($r_t^{m,n} \in R$, the set of all requests). Requests should describe everything agent m desires of agent n at time t. They are used both for direction and describing desired changes to common goals. Our model makes no specific commitment to representation; however, we expect that goal reasoning agents might directly exchange lists of goals, preferences, and constraints.

Explanations are intended to communicate information about an agent's internal state that motivates that agent's current behavior (e.g., "I moved the box because it was blocking my vision", "My battery is low so my movement range is limited"). Each teammate m provides an *explanation* $x_t^{m,n}$ to each other teammate n ($x_t^{m,n} \in X$, the set of all explanations). These explanations should help other teammates to understand an agent's actions and predict their future actions, to facilitate coordination. One particular area of importance is that an agent should explain why it does or does not pursue another agent's request; if an agent does not, for example, have sufficient resources to succeed, this may prompt the requester to provide resources or assistance.

Note that the explanations described here are proactive and not query-based. While query-based explanations are an important problem, a clean separation of agent-based coordination and decision-making issues from natural-language issues will permit objective evaluations and comparisons without human interaction issues. We expect, however, that an external query interface could be provided that translates queries into informational requests.

Each teammate m uses the various pieces of information they have received over time[1] (i.e., observed environment states, received requests, and received explanations) along with their sent requests (and, implicitly, their internal motivations) to guide their action selection policy $\pi^m : O^{|T|} \times R^{|M|} \times X^{|M| \times |T|} \times R^{|M|} \to A$. This policy is expected to be dynamic, and may be influenced by an agent's interactions with its teammates, as well as by the environment. A typical goal reasoning agent's policy may involve considering and reselecting goals and replanning to achieve them, but the model accommodates various types of policies.

We also model the *satisfaction* of each teammate, which describes how well an agent's desires are being met. Satisfaction is a function of an agent's observations (which may indicate the achievement of desired states), requests made and received (which help determine the success and failure of collaboration), and explanations received (which may justify failures or provide confidence in the current collaboration): $sat^m : O^{|T|} \times R^{|M|} \times X^{|M| \times |T|} \times R^{|M|} \to \mathbb{R}$. The satisfaction of the entire team can also be modelled as a function of each teammate's satisfaction ($f(sat^1, \ldots, sat^{|M|})$); optimizing this measure incorporates an agent's own satisfaction, as well as the estimated satisfaction of each of its $|M| - 1$ teammates.

To exemplify how our model could be used in practice, we describe it in terms of four variations on the human-agent teaming problem that describe existing goal reasoning work: *single supervisor*, *silent teammates*, *silent assistant*, and *rebel agent*. These examples are not meant to be exhaustive, but instead to show that our model can represent common team structures encountered in goal reasoning research.

2.1 Single Supervisor

Even autonomous goal reasoning agents often receive goals or tasks from an outside source. In this framework, we model that source as an agent who makes requests and wants explanations to understand what the agent is doing to fulfill them. This results in the Single Supervisor version of the human-agent teaming problem model, shown in Figure 2. In this version, an agent has a single teammate whose satisfaction it wishes to maximize, referred to as the *supervisor*. While both teammates can sense and act in the environment[2], the superior-subordinate relationship results in requests and explanations being unidirectional (i.e., the agent cannot make requests of the supervisor and the supervisor does not explain itself to the agent). As such, the agent's action selection policy does not include explanations it has received or

[1] We assume that, since the requests at the current time contain the complete request to/from each agent, the policy does not need to consider past requests. If this is not the case, the action selection policy can be extended to include past requests.

[2] Although the supervisor does not need to be situated in the environment.

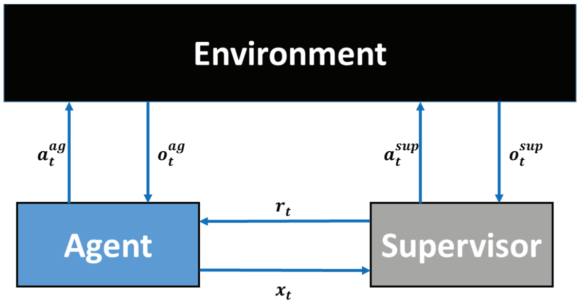

Figure 2: Single Supervisor version of the Human-Agent Teaming Problem Model

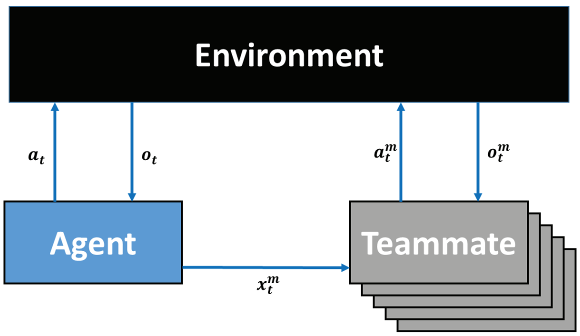

Figure 3: Silent Teammates version of the Human-Agent Teaming Problem Model

requests it has sent, and only deals with a single teammate (the policy is simplified to $\pi : O^{|T|} \times R \rightarrow A$). The primary performance measure for this problem is the supervisor's true satisfaction, measured either at the termination of interaction, or as an average over time.

2.2 Silent Teammates

In the Silent Teammates version of the human-agent teaming problem model (Figure 3), an agent operates as a member of a human-agent team, but does not receive any direct requests from its teammates. This is an unusual teaming arrangement, but necessary when a team is communication-restricted in some way (possibly to avoid giving an adversary knowledge). In this problem, the agent does not make requests of other teammates, nor expect explanations from them. However, the agent still provides an explanation on demand, to assist teammates in understanding when they have questions. An example of such a goal reasoning agent is the Autonomous Squad Member (ASM), an agent controlling an unmanned ground vehicle that is embedded in a team of humans (Gillespie et al. 2015). The ASM agent must infer and respond to teammates' desires (e.g., follow along, provide cover in a fight) without explicit requests. This results in an action selection policy that inputs only observations: $\pi : O^{|T|} \rightarrow A$. Similarly, the satisfaction function does not include requests: $sat^m : O^{|T|} \times X^{|M| \times |T|} \rightarrow \mathbb{R}$. The primary performance measure in this problem is the team's overall satisfaction.

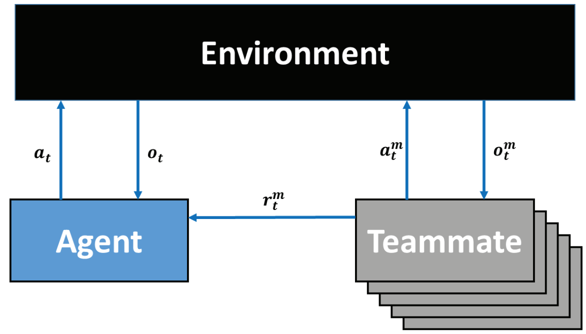

Figure 4: Silent Assistant version of the Human-Agent Teaming Problem Model

2.3 Silent Assistant

The Silent Assistant version is a multi-agent teaming problem with no explanation requirement (Figure 4). In this example, the agent assists one or more other agents by acting on their requests, but does not provide explanations, receive explanations, or make requests of others (i.e., it does not initiate coordination). An example of such a goal reasoning agent is the Tactical Battle Manager (TBM), an agent that controls an unmanned air vehicle while serving as a wingman for an aircraft controlled by a human pilot (Floyd et al. 2017). The TBM operates autonomously but receives explicit tasks from a human pilot. The lack of communication from the agent is largely due to the real-time adversarial nature of the domain; goal changes are motivated by dangerous situations or opportunistic targets, so explanations are not a primary requirement for this system. Additionally, since the TBM is a human pilot's wingman, it serves a subordinate role and therefore does not generate requests. As such, the agent's action selection function and the satisfaction functions do not include explanations or requests from the agent ($\pi : O^{|T|} \times R^{|M|} \rightarrow A$, $sat^m : O^{|T|} \times R^{|M|} \rightarrow \mathbb{R}$). The primary performance measure is the team satisfaction function.

2.4 Rebel Agent

The previous three problem versions we described assume that the agent's primary drive is to satisfy teammates' requests. In the Rebel Agent version (Coman, Gillespie, and Muñoz-Avila 2015), an agent has internal goals or motivations that differ from (and may conflict with) those of its teammates. There are two ways in which a rebel agent can be represented using our model. The simplest method is to consider the agent as a member of its team but having internal motivations that are unknown to its teammates. Thus, when attempting to maximize team satisfaction it may prioritize its own satisfaction above the satisfaction of its teammates (e.g., provide them with different weights). The ARTUE agent (Molineaux, Klenk, and Aha 2010) is a rebel agent that receives explicit requests in the form of goals that it may choose to ignore in order to achieve goals more important to it. A more complex representation would be to consider the agent to be a member of two teams concurrently (i.e., in Figure 1 the agent would be at the intersection of two teams). For example, consider an agent that is a member of a *corporate catering team*, but is also a member of a *vegetarian team*. While the agent contributes toward achieving catering goals (e.g., host a successful event, maximize profit) it may choose actions to maximize the vegetarian team's satisfaction (e.g., minimize the amount of meat used). In the internally motivated case, the primary performance measure is a team/rebel satisfaction function $f(sat^1, \ldots, sat^{|M|}, mot(s_{final}))$, where $mot(s_{final})$ describes how well a rebel's internal motivations are satisfied in the true final state of the environment. In the dual-membership case, the primary performance measure is a combined function of two (or potentially more) team satisfaction functions: $f_C(f_1(sat^1, \ldots, sat^{|M|}), f_2(sat^1, \ldots, sat^{|M|}))$.

2.5 Assumptions

Consideration of important assumptions is necessary for this framework. Existence of the transition and observation functions means that environments can be static or dynamic, deterministic or probabilistic, and fully or partially observable. Existence of the policy and satisfaction functions of teammates implies that we should also consider whether to assume complete or incomplete knowledge about these functions, and whether information given regarding them (i.e., requests and explanations) is perfect or noisy. This cuts across all problems, and those purporting to address these problems should state their assumptions regarding these functions.

3 Examples

Requests and explanations can take many forms including natural language utterances, structured text, or low-level state representations. In this section we provide examples of requests, explanations, and how they can be used.

Requests: In general, we expect requests to vary in complexity across agents. An example complex request representation might be a tuple $\langle S_{avoid}, F_{prefs}, G, C\rangle$, including constraints $S_{avoid} \subset S$ in the form of states to avoid (e.g., "battery should never fall below 10%"), preference functions $F_{prefs} : S \times S \rightarrow \{True, False\}$ (e.g., "spend as little money as possible"), goal states $G \subset S$ (with or without priorities), and context C that describes why achievement of a particular goal is desired (e.g., the reason for requesting an agent to cook food could be because (1) 'supervisor is hungry' or (2) 'supervisor needs to bring food to a dinner party later'). Context and preferences are especially relevant for goal reasoning agents, as these can guide which goals should be considered when goal change is warranted. Additionally, the reasons for a supervisor's request of a goal are likely to be useful in making goal change decisions; for example, the context may include a higher-level goal of which the current request is a subgoal (e.g., a "cook food" goal is a subgoal of a ¬hungry goal).

Explanations: An important reason for explanations is that goal reasoning agents may change their local objectives (i.e., subgoals) in response to changes in the environment preventing the accomplishment of the original task. Thus, whenever an agent changes its goal, an explanation could be a tuple $\langle g_{failed}, c_{failed}, g_{new}, p_{new}\rangle$ composed of a failed goal g_{failed}, description of state properties that prevent goal achievement c_{failed}, new goal g_{new}, and new plan p_{new}. Note that in this framework, explanations are always proactive for simplicity of discussion; to support reactive explanations, an external interface could store this information to present to a human in answer to specific queries.

A Supervisor Requests Cake: We now describe an example of the Single Supervisor problem: first, a human supervisor σ makes a request of a chef agent α to "*bake me a chocolate cake that I can eat when I get home*". Here, the request $r_t^{\sigma,\alpha}$ is the tuple $\langle\emptyset, \emptyset, \{\{$*exists(chocolate-cake), on(chocolate-cake,table)*$\}\}, \{$*hungry(me), wants(me, chocolate)*$\}\rangle$, which describes a single goal state based on the original English utterance (translating human utterances to goals has garnered attention in the human-robot interaction community, see (Briggs, McConnell, and Scheutz 2015) for an example). No constraints or preferences are provided.

The chef agent α represents its supervisor's satisfaction function sat^σ as a weighted average of (1) the percentage of his desires that are satisfied in the current state and (2) the time delay between t (time of request issuance) and t_a (time of request achievement). Based on this, the agent uses an automated planner to produce a plan that achieves the requested goal in the shortest possible time. Its policy π^α removes the first action from this plan and executes it; this is repeated until the following action a_t^α is known to be inadmissible based on a state observation o_t^α. We now describe a situation that may warrant the agent to consider goal change.

Soon after it begins acting to achieve the goal, the agent discovers it cannot continue baking because there is no cake flour in the kitchen. The agent considers adoption of a new goal *acquire(cake-flour)*, and creates a plan: go to the grocery store, purchase cake flour, and return. However, the plan to accomplish the new goal would significantly increase the time required to fulfill the supervisor's request. Knowing that the supervisor is hungry and wants chocolate cake, the chef agent decides to instead switch to a goal to make chocolate chip pancakes, which seems like a reasonable substitute. When the supervisor comes home, the agent provides him with an explanation:

$\langle\{$*exists(chocolate-cake), on(chocolate-cake, table)*$\}$,
$\{$*available(cake-flour)*$\}$,
$\{$*exists(pancakes), on(pancakes, table)*$\}$,
$\{$*acquire(pancake-mix), acquire(chocolate-chips), bake(pancakes, pancake-mix, chocolate-chips), serve(pancakes)*$\}\rangle$.

This explanation serves to communicate why the agent changed its goal, and what it did instead. If the context of the supervisor's request had been a birthday party, the agent α might have reasoned that the subgoal of going to the grocery store was warranted.

In general, the issue of how much information must be exchanged between teammates is unresolved. In this example, we assume sufficient knowledge to minimize the need for communication; for example, the agent knows that the supervisor's desires would be met to some degree by chocolate chip pancakes. Future work on goal reasoning agents will need to consider this question.

4 Evaluating Explainable Goal Reasoning Agents

We expect that typical evaluations will consider a specific problem and assumptions, and show results on a primary performance metric in a subset of domains. Results should be directly comparable with other agents that make the same assumptions, use the same domain, and use a similar set of teammates. For this reason, sharing domains as well as appropriate automated teammates (i.e., other software agents that are part of the team) should promote comparison.

When discussing the four versions of the human-agent teaming problem, we briefly described the various metrics that can be used to measure whether the goal reasoning agent is an effective member of the team. However, in addition to agent performance there is also the issue of how well the agent interacts with its human teammates. In these cases, evaluations should consider whether the provided explanations are appropriate for aiding human collaboration. We consider metrics for explanation as falling into four categories: *tests of explanation quality*, *tests of user satisfaction*, *tests of user comprehension*, and *tests of user or user-system team performance*. These are based directly on Hoffman, Klein and Mueller's (2017) work on evaluating explanations. Two agents need not use the same explanation representation (e.g., natural language, internal state variables) to be compared.

Tests of Explanation Quality: Experiments that measure explanation quality can be conducted without humans in the loop, but often still require a human to assess the results. These can be compared against explanations generated by another system or by a human. Some measures of explanation quality are surveyed in Table 1.

Tests of User Satisfaction: These should solicit a user's subjective satisfaction with an agent's performance, typically using Likert scale questions.

Tests of User Comprehension: These gauge how well explanations generated by an agent improve the accuracy of a user's mental model of an agent's behavior. For explainable autonomous agents, experiments could include questions about the system's policy to measure user understanding.

Tests of User or User-System Team Performance: These measure how explanation affects the user's ability to accomplish some task, often an interactive task involving the explaining agent. A scenario-specific performance metric can be used to evaluate the team's performance for this purpose. To provide a comparison, the same evaluation should be applied with and without agent-provided explanations, and, if possible, against a human-only team.

Table 1: Abstract Measures of Explanation Quality

Soundness	Plausibility, internal consistency
Appropriate Detail	Amount of detail and its focus points
Veridicality	Does not contradict the ideal model (although there are times when inaccurate explanations work better for some users and some purposes)
Usefulness	Fidelity to the designer's or user's goal for system use
Clarity	Understandability
Completeness	Relative to an ideal model
Observability	Explains an agent mechanism
Dimensions of Variation	Reveals boundary conditions

5 Conclusions and Future Work

We have presented new formal models and problem variations for human-agent teaming, in hopes of promoting comparisons, competitions, and sharing of evaluation code among goal reasoning researchers. We have made the case that explanation is an important and attainable capability for goal reasoning agents. Finally, we have described useful evaluations to be used to provide evidence of how well both goal reasoning agents and human-agent teams, perform.

In future work, we will produce refined models based on community feedback; furthermore, we will provide concrete problem instances and representations for use in benchmarking and comparison.

6 Acknowledgements

This research was developed with funding from DARPA. The views, opinions, and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the DoD or the U.S. Government. Table 1 is used with permission from the Institute for Human and Machine Cognition and was created by IHMC, MacroCognition LLC, and Michigan Technical University, with the support of the DARPA XAI Program DARPA-BAA-16-53.

References

Bonnano, D.; Roberts, M.; Smith, L.; and Aha, D. 2016. Selecting subgoals using deep learning in Minecraft: A preliminary report. In *Working Notes of the IJCAI-16 Workshop on Deep Learning and Artificial Intelligence*.

Briggs, G.; McConnell, I.; and Scheutz, M. 2015. When robots object: Evidence for the utility of verbal, but not necessarily spoken protest. In *International Conference on Social Robotics*, 83–92. Springer.

Coman, A.; Gillespie, K.; and Muñoz-Avila, H. 2015. Case-based local and global percept processing for rebel agents. In *Case-Based Agents: Papers from the ICCBR 2015 Workshops*, 23–32.

Cox, M. T. 2013. Goal-driven autonomy and question-based problem recognition. In *Proceedings of the Second Annual Conference on Advances in Cognitive Systems, Poster Collection*, 29–45.

Dannenhauer, D., and Muñoz-Avila, H. 2015. Goal-driven autonomy with semantically-annotated hierarchical cases.

In *Proceedings of the 23rd International Conference on Case-Based Reasoning*, 88–103.

Floyd, M. W.; Karneeb, J.; Moore, P.; and Aha, D. W. 2017. A goal reasoning agent for controlling UAVs in beyond-visual-range air combat. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 4714–4721. AAAI Press.

Geib, C.; Weerasinghe, J.; Matskevich, S.; Kantharaju, P.; Craenen, B.; and Petrick, R. P. 2016. Building helpful virtual agents using plan recognition and planning. In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Gillespie, K.; Molineaux, M.; Floyd, M. W.; Vattam, S. S.; and Aha, D. W. 2015. Goal reasoning for an autonomous squad member. In *Goal Reasoning: Papers from the ACS 2015 Workshops*, 52–67.

Golpayegani, F., and Clarke, S. 2016. Goal-based multi-agent collaboration community formation: A conceptual model. In *Workshop on Goal Reasoning at IJCAI-2016*.

Hoffman, R.; Klein, G.; and Mueller, S. 2017. Initial concepts and literature review for DARPA-XAI. Technical report from task area 2 (psychological model of explanation) on DARPA contract DARPA-BAA-16-53. Technical report, Institute for Human and Machine Cognition, Pensacola, FL.

Jaidee, U.; Muñoz-Avila, H.; and Aha, D. W. 2013. Case-based goal-driven coordination of multiple learning agents. In *Proceedings of the 21st International Conference on Case-Based Reasoning*, 164–178.

Klein, G.; Woods, D. D.; Bradshaw, J. M.; Hoffman, R. R.; and Feltovich, P. J. 2004. Ten challenges for making automation a ”team player” in joint human-agent activity. *IEEE Intelligent Systems* 19(6):91–95.

Molineaux, M.; Klenk, M.; and Aha, D. 2010. Goal-driven autonomy in a Navy strategy simulation. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 1548–1554.

Richards, D., and Stedmon, A. 2017. Designing for human–agent collectives: display considerations. *Cognition, Technology & Work* 19(2-3):251–261.

Roberts, M.; Vattam, S.; Alford, R.; Auslander, B.; Karneeb, J.; Molineaux, M.; Apker, T.; Wilson, M.; McMahon, J.; and Aha, D. W. 2014. Iterative goal refinement for robotics. In *Planning and Robotics: Papers from the ICAPS Workshop*. AAAI Press.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*. MIT Press Cambridge.

Weber, B. G.; Mateas, M.; and Jhala, A. 2010. Applying goal-driven autonomy to starcraft. In *Proceedings of the Sixth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.