Advances in Cognitive Systems 5 (2018) 97-111 Submitted 8/2017; published 5/2018

Improving the Robustness of Team Collaboration through Analysis of Qualitative Interactions

Matthew Klenk KLENK@PARC.COM
Daniel G. Bobrow BOBROW@PARC.COM
Johan de Kleer DEKLEER@PARC.COM
Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304 USA

Wendy Mungovan WENDY.MUNGOVAN@BARNSTORMRESEARCH.COM
Jorge Tierno JORGE.TIERNO@BARNSTORMRESEARCH.COM
Barnstorm Research, 745 Atlantic Avenue, 8th Floor, Boston, MA 02111 USA

Abstract

Members of effective teams must have knowledge about each other's future actions. Typically, this is done through messages or precomputed divisions of labor. The former requires ongoing communication between the agents and the latter constrains the autonomy of the individual agents. We introduce *coordination rules* that facilitate collaboration between autonomous agents when communication is lost. By envisioning the results of all possible plan executions for each agent, we identify which actions result in the greatest increase of within-team uncertainty. If removing this action does not substantially reduce the expected utility of the plan, we create a coordination rule, a statement that the agent will or will not take a particular action in some possible future. Coordination rules facilitate collaboration by improving prediction by teammates. Towards supplying cognitive systems with coordination rules, this paper makes three contributions. First, we identify qualitative interactions by representing the space of decisions available to the agents in *plan-with-options* and their consequences in a *factored envisionment* that compactly represents multi-agent simulations. Second, we define two classes of within-team uncertainty metrics with respect to the envisionment. Third, we present an evaluation of the effects of coordination rules on action selection in three scenarios. In all three scenarios, coordination rules enabled extended planning horizons and reduced planning times with no substantial effect on plan quality.

1. Introduction

Coordinated goal-oriented behavior is a hallmark of intelligence. Groups that need to coordinate actions include trained teams (e.g., athletes executing a play), familiar novices (e.g., a family arranging childcare and meals), or complete strangers (e.g., drivers navigating an intersection). Deciding what to do is difficult in these scenarios due to ambiguity introduced by other agents' actions. Coordination can be achieved by constraining autonomy (e.g., athletes executing a play), division of labor (e.g., dad cooks dinner while mom picks up the kids), and social norms (e.g., at a four-way stop, the car on the right gets to go first). In new or changing situations, agents coordinate by considering other agents' goals and reasoning about possible futures. This preserves the ability of the agents to respond autonomously to failures and opportunities that arise during plan execution while still coordinating their behavior, even in the absence of communication. We define *coordination rules* as restrictions to the space of actions available to individual agents during execution. We demonstrate that, when communication is not possible, coordination rules facilitate collaboration by reducing within-team uncertainty.

To enable coordinated behavior, it is necessary to take into account the uncertainty of the dynamics of the world as well as the decisions of other agents. We use a standard planning model to represent the dynamics of the world and introduce *plan-with-options* as a model of contingent plans to represent the space of decisions left to the agent. *Envisioning* from qualitative reasoning (Kuipers, 1994; Weld & de Kleer, 1989) is a multi-trajectory simulation process that given a model analyzes all qualitatively distinct futures for a scenario. In the multi-agent setting, many of the distinctions captured by traditional approaches are not relevant. Therefore, we introduce *interaction-based factoring* that creates a compact representation of possible futures. To automatically construct coordination rules, we define two classes of within-team uncertainty metrics and how they are computed over the envisionment. Finally, we evaluate these ideas over a set of scenarios.

The primary contributions of the research that are described in this paper include:

- Definition of coordination rules as constraints on agents' actions to facilitate collaboration
- Exposition of an interaction-based factoring algorithm that produces a compact representation of a multi-agent envisionment
- Description and discussion of two classes of within-team uncertainty metrics
- Evaluation of the effects of coordination rules on action selection in three scenarios.

This is a cognitive systems problem (Langley, 2012) because it focuses on agents representing and reasoning about other agents' behaviors within a planning and execution system. Furthermore, our contributions combine representations and algorithms from artificial intelligence (AI) planning and qualitative reasoning. Finally, we show how an automated analysis of qualitative interactions can facilitate collaborative behavior, an essential trait of cognitive systems.

2. Representing Plans and Futures

Our scenarios involve agents with different capabilities, team-level evaluation, and uncertainty in our own actions, teammates' decisions, and adversary intentions. The Planning Domain Definition Language (PDDL) provides a standard mechanism for describing the dynamics of the world and scenarios (McDermott et al., 1998). We use PDDL to represent the preconditions and effects of actions. Uncertain effects are modeled using multiple actions with the same name and preconditions but different effects. Table 1 contains a PDDL example for an Unmanned Aerial Vehicle (UAV) successfully destroying a target. These actions may be annotated with associated probabilities.[1]

An *envisionment* (Weld & de Kleer, 1989) is a graph of qualitative states with edges indicating the possible successors. When changes are the result of actions, an envisionment is analogous to

1. Probabilistic PDDL (Younes & Littman, 2004) would perhaps be a cleaner representation, but it is not as well supported by existing planners.

Table 1. PDDL example for an action of a UAV firing a missile at a target and destroying it.

```
(:action fire-success
   :parameters (?uav - uav ?target - sam )
   :precondition (and (in-range ?uav ?target)
                      (alive ?target)
                      (alive ?uav)
                      (> (weapons ?uav) 0)
   :effect (and (not (alive ?target))
                (destroyed ?target)
                (decrease (weapons ?uav) 1))
```

the state-action space of planning (Ghallab et al., 2004). States are defined by the collection of propositions that are true, and edges in the graph are labeled by the action (or set of actions, in the multi-agent case) that changes the state.

Consider a scenario with two rovers, a survey rover and an extraction rover, that are collaborating to extract minerals (shown in Figure 1). The world has a base location and four possible mining locations. The survey rover can move between any of the locations and use its sensors to identify if minerals are present. The extraction rover can also move to any location and use its drill. There is also a joint action where the survey rover uses its sensors to guide the extraction rover's drilling action. The full envisionment has 200 states.

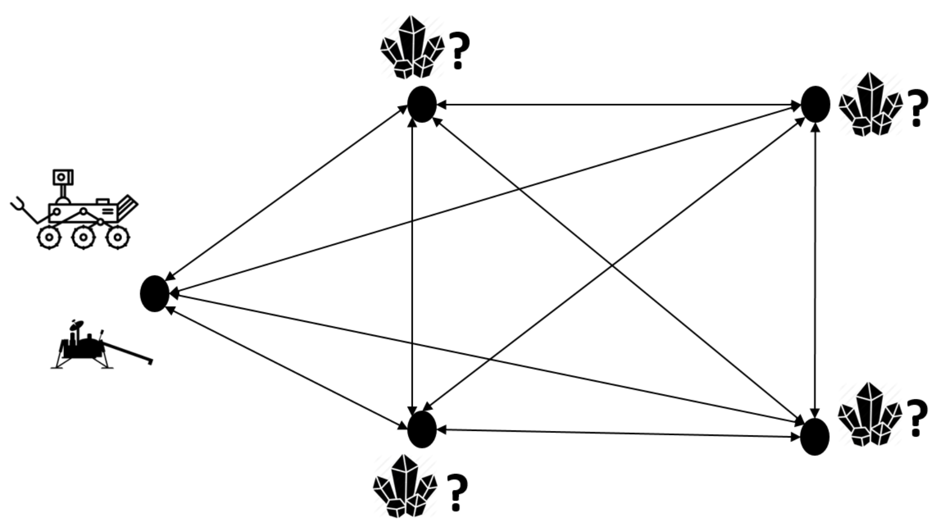

Figure 1. Collaborative mining domain using two rovers. The rovers can move between the locations, the survey rover can sense for minerals at a location, the extraction rover can extract minerals at a location, and if both rovers are at the same location, they can perform a joint action to extract minerals. Given the increased precision of the sensors, this joint action has a higher yield than the extraction rover working alone.

As in AI planning, the number of states in the envisionment is, in the worst case, exponential in the number of ground predicates in the scenario. This is because each possible state is a truth assignment to each of the ground predicates. Therefore, we introduce two concepts that facilitate the representation of large scenarios: (1) plan-with-options, which describes the actions available for teammates in collaborative actions, and (2) interaction-based factoring, which enables reasoning about possible futures.

2.1 Plan-with-Options

One cause for the large state-action space is that every afforded action will result in a possible state transition. To reduce this space, we want to consider their decision process, and to accomplish this, we need to have an explicit representation of the options they would consider.

In the rover example, plan-with-options can be used to constrain the search pattern of the agents. For example, the survey rover may only visit the locations in a particular sequence with the option of returning home after visiting each one, while the drilling rover must follow the same pattern and only drill when colocated with the survey rover. Instead of having to generate successor states from all pairs of rover locations for the extraction rover's drilling action, we only generate successor states for the drilling action from states in which the rovers are co-located.

We represent these choices using hierarchical task networks (HTNs) (Erol et al., 1994). These represent tasks hierarchically and HTN planning involves decomposing tasks into subtasks and finally primitive actions that can be executed by the agent. An HTN *method* specifies the decomposition of a task. If multiple methods apply to the same task, then there are multiple plans that satisfy the task. We define a *plan-with-options* to be all possible decompositions for a goal task, as well as the current sequence of actions each agent is pursuing. Therefore, instead of reasoning over every possible action available to every agent, we consider the space of decomposition decisions faced by each agent. This provides the agent autonomy to respond to changing conditions while remaining committed to the collaborative activity.

To implement plan-with-options, for each scenario, we define a hierarchical task network for each agent. We model options with multiple decompositions for the same task. For this work, our agent uses SAPA (Do & Kambhampati, 2003), a domain-independent planner that generates all possible sequences of states. We then translate the HTN into PDDL notation using Alford et al.'s (2009) approach of introducing new atoms to restrict the range of applicable actions based on method preconditions. In future implementations of these ideas, we intend to use the HTN planner SHOP2 (Nau et al., 2003) instead.

2.2 Interaction-Based Factoring

The other main driver in the complexity of the envisionment is the explicit representation of temporal differences between agents' actions. If the survey rover is headed toward location 1 and the extraction rover is headed toward location 2, standard representations would include six states:

1. Both rovers en route to their respective locations
2. Survey rover at location 1 and extraction rover en route to location 2
3. Survey rover sensing at location 1 and extraction rover en route to location 2

4. Survey rover en route to location 1 and extraction rover at location 2
5. Survey rover at location 1 and extraction rover at location 2
6. Survey rover sensing at location 1 and extraction rover at location 2.

It does not matter which rover arrives first, unless they are interacting with each other from their respective locations. Therefore it should be possible to represent the survey rover's actions with three states (en route, at location 1, and sensing at location 1) and the extraction rover's action as two states (en route and at location 2). While this appears to only remove a single state, consider what happens when we allow the drilling rover to drill when it reaches its destination. In the baseline case, there will be three additional states (one for 4, 5, and 6), but, when considered separately, there would only be a single additional state.

Model decomposition has a long history to improve scaling of qualitative simulations. For example, DecSIM separates the envisionment into components with limited interactions (Clancy & Kuipers, 1997). This idea was extended to battlespace planning by analyzing the spatio-temporal trajectories of each agent (Hinrichs et al., 2011). We extend this idea with *interaction-based factoring*, in which decisions about when agents are reasoned about individually versus as a group are made using the planning domain model. The core idea is that if an action includes multiple agents in its arguments, then those agents are *interacting* in that action. Factored envisionments consist of *factors*, which are envisionments of a subset of the agents and *factored states* that exclude facts related to other agents.

Interaction-based factoring works in two stages, both of which operate recursively. In the first stage, the system identifies subsets of agents that are able to take actions together in the current state. That is, there exists an action for which the preconditions are met and each of the agents is a participant. For each subset of agents, including the original set, the system creates a factored state, which includes the facts related to those agents and objects in the world. From this factored state, the system creates an envisionment using all the actions in which the subset of agents are interacting. This envisionment is a factor. For each state in each factor, if further subsets of agents from that factor are afforded actions, the system creates additional factors from each of those factored states. In the second stage, the system identifies sets of factored states from across the factors of the previous stage called *join state sets*, which include a single factored state from each factor being joined. Each join state set results in a new factored state if there are new actions that can be taken by the set of agents in the join state set. This recursive factoring and joining approach continues until there are no further actions that result in new factors to be created.

Consider our rover example again. The envisionment includes three sets of agents: one for each rover and one for the rovers together. A single trajectory will involve splitting from the initial state as both rovers move to different locations, joining when they collaborate to drill the minerals, and separating again as they travel back to home. The complexity due to the locations of the rovers alone goes from 25 states (one for every combination of rover locations) to 15 states in the factored envisionment (five locations for the survey rover factor, five locations for the extraction rover factor, and five states for each co-location). In the full envisionment, the state changes for sensing and drilling actions are multiplied by 25, but in the factored envisionment, the sensing actions are in the survey rover factor and the drilling actions occur in the extraction rover factor. This results in an exponential reduction in qualitative states in the factored envisionment.

As shown in the example, there are substantial savings when moving to the factored envisionment. Theoretically, if the state-action space of two individual agents are M and N, then their combined envisionment would include $M \times N$ states. If there was no interaction, their factored envisionment would include only $M + N$ states. If they were interacting in every state, then the factored envisionment would still be on the order of $M \times N$ states. Most collaborative problems have a mixture of interaction and independence. To evaluate the impact of factoring on the size of the state graph, we created a series of five scenarios with increasing complexity. These scenarios involved a set of aircraft striking a set of defended targets. Scenarios were made more complex by adding additional aircraft, locations, and targets. In Problem 1, a single UAV strikes a single target with two locations. In Problem 2, two UAVs strike two colocated targets. In Problem 5, there are additional locations for ingress and egress rendezvous, three targets are at different locations, and the UAVs are expected to attack remaining targets until either the UAVs or targets are destroyed or the UAVs are out of weapons.

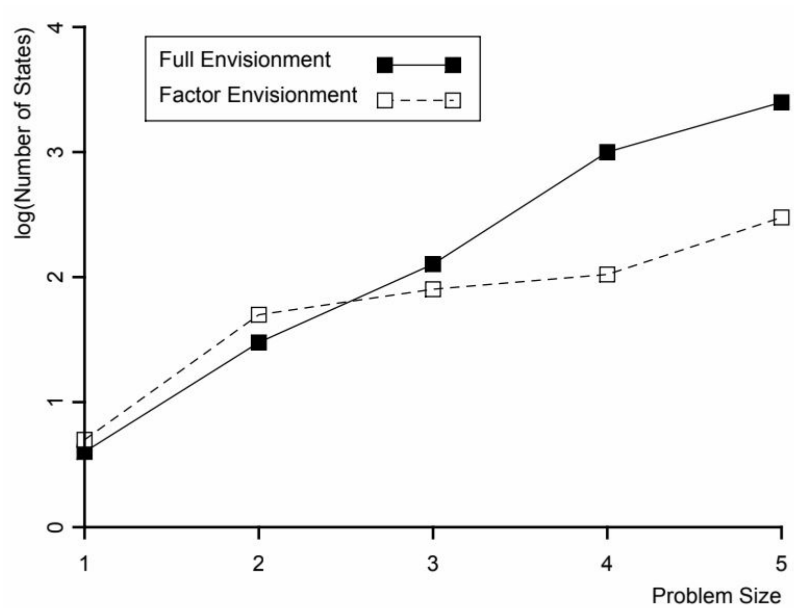

Figure 2. The effects of factoring on the size of the envisionment in air strike scenarios.

Figure 2 plots the problem complexity against the number of states. In the smaller scenarios, all of the agents are constantly interacting and the factored envisionment is slightly larger. This is due to the overhead of creating initial states for each factor. As the problems get more complex, reasoning about non-interacting agents independently produces an order-of-magnitude savings. Through plan-with-options and interaction-based factoring, we compactly represent and reason about the possible future states of collaborators. In the next section, we discuss how we generate coordination rules from envisionments.

3. Generating Coordination Rules

Coordination rules reduce within-team uncertainty by restricting the space of possible actions for a particular agent. By analyzing the envisionment, the agent measures the amount of uncertainty introduced by each possible coordination rule of its decisions. Next, it generates a coordination rule that eliminates the action with the greatest increase in within-team uncertainty in which the remaining decisions still support a successful collaboration. When action probabilities and utility functions are known, it is possible to evaluate the tradeoff between within-team uncertainty and overall expected utility.

To generate coordination rules, it is necessary to have a model the other agents. We assume that other agents observe the environment and select actions. While the envisionment captures the space of their possible actions and their outcomes, it does not include any model of their decision-making process. Without this, we assume that each action and outcome is equally likely. Given probabilistic action models and goals, we model the other agents as utility maximizing. Uncertainty in the other agents' observations may be captured in the envisionment (e.g., detection events are modeled as agent actions). Because we cannot ensure future communication, future states of the world are uncertain (e.g., one teammate will not know what the other just did). Therefore, even with a utility maximization model for other agents' actions, there is still uncertainty about what other actions they will take. With or without probabilities, the envisionment captures the space of possible actions for all of the agents. To define within-team uncertainty from an envisionment, we must to answer two questions: *what aspects of the world are important* and *when do we want to measure uncertainty?*

To answer the first question, we identified two types of metrics. *State-based* uncertainty metrics calculate the uncertainty between the values of certain predicates in possible future states of the world. In contrast, *action-based* metrics measure the uncertainty between possible action sequences for each agent. For either metric, it is necessary to determine the time period of interest. For example, when deciding which items to pickup in a grocery store, it is helpful to know what part of the store your partner will be in for the next few minutes, as it is likely you will run into each other soon. On the other hand, if you are working on a document without communication, you care about which sections of the document your colleague may have changed by the next possible communication point.

Consider a strike scenario in which two UAV's are attacking two targets. We use three features to create a state-based entropy measure: the location of the UAV, the state of its targets, and the status of its weapons. We create a unique class for each combination of feature values. To measure the uncertainty at the end of the mission, we assign each end state to its respective class. The distribution of these classes provides a measure of *entropy* (Shannon, 1948). Coordination rules operate by ruling out individual actions. For each action, we remove all successor states that require it. From the remaining states, we calculate entropy in the same manner. To create the coordination rule, we select the action resulting in the largest entropy reduction that still allows for successful plan execution. In this case, UAV 1 tells UAV 2 that if communication is lost, it will not pursue Target 2 while enroute to Target 1.

To better understand the applicability of these types of metrics, we used three scenarios. For each scenario, we identified the coordination rule that best captures our intuitions. Then we deter-

Table 2. Results for the selection of the expected coordination rule by each metric.

Scenario	State	Action
SEAD	✓	✗
Patrol	✗	✓
Strike	✓	✓

mined if each metric scored each expected coordination rule the highest. For the state-based metric, we used the terminal states of the envisionment. For the action-based metric, we used the complete action trajectories. Here are the three scenarios, along with the expected coordination rules:

- Suppression of Enemy Air Defenses (SEAD): Two aircraft must search two regions and destroy enemy air defenses. In addition to moving actions, there are sensing and attacking actions for both friendly and enemy units.
  - Expected Rule: Aircraft 1 visits target region 1 before target region 2.
- Patrol: Two aircraft with different sensors must identify and track targets within an area of interest. Collaboration is required for the search aircraft to hand off targets to the tracking aircraft.
  - Expected Rule: Search aircraft will only follow targets within one of the regions on the return trip.
- Missile Strike: Two salvos of missiles are launched at high-priority and low-priority targets. The targets have countermeasures that may jam the communications or destroy the salvos, and the salvos can change what they are targeting.
  - Expected Rule: If each salvo was alive at the point of communication loss, the first salvo that was targeting the high-priority target will engage the low-priority target.[2]

Table 2 indicates whether each metric was able to identify the expected coordination rule. State-based uncertainty metrics appear to be most effective when there are clear aspects of the state that are relevant to mission decisions (e.g., the status of a particular target). Action-based metrics seem to work best in scenarios with repeated patterns of activity (e.g., moving between different regions in a patrol).

In this section, we identified two important questions that define a space of possible methods for generating coordination rules. We evaluated two design choices within this space by comparing the rules they generate with expected rules for three scenarios. In the next section, we assess how coordination rules effect multi-agent collaboration.

2. This is due to the higher likelihood that the first salvo will be destroyed.

4. Impact on Action Selection

We claim that coordination rules support autonomous collaboration by facilitating action selection of teammates that cannot directly communicate with one another. This collaboration could include human agents or artificial agents with differing action-selection mechanisms. We can formalize this problem using decentralized partially observable Markov decision processes (Dec-POMDP) (Bernstein et al., 2000). While we are not arguing that people use this abstract representation, the Dec-POMDP framework provides an established formalism to measure the impact of coordination rules on the collaboration problem.

Coordination rules result in a smaller state-action space, and therefore we expect off-the-shelf solvers to create plans faster and with longer horizons after applying coordination rules. Given that such rules remove options, they may reduce the quality of the optimal solution. On the other hand, they may also improve the value of approximate solutions by simplifying the action-selection process. We explored these issues over another three scenarios. The independent variable is the presence or absence of the coordination rule. If there is a coordination rule, we modify the Dec-POMDP for the problem. Next, we apply standard Dec-POMDP solvers and measure three dependent variables: the time to solution, solution quality, and maximum planning horizon.

4.1 Decentralized Partially Observable Markov Decision Process

Decision-theoretic planning represents the decision problem faced by agents by modeling their sensors, actions, environment, and task. Extending this to the multi-agent setting, as needed for collaboration problems, requires integrating game-theoretic models. Dec-POMDP is one such extension for scenarios with joint payoff but decentralized actions and observations. As this property aligns with the scenarios described in this paper, we selected this framework to explore the impacts of coordination rules on the collaborative problem-solving.

A Dec-POMDP is defined by a set of agents, a set of states (including an initial state distribution), a set of actions for each agent, a transition function that specifies the probability of each state given a set of actions taken by the agents, a reward function that specifies the reward for being in a state, the set of observations for each agent, and an observation model that specifies the probability of each observation for each agent given a state. A solution is a set of policies, one per agent, which maps each sequence of observations to an action for each agent. In the worst case, this is NEXP-complete (Bernstein et al., 2000). While we do not expect cognitive systems to employ Dec-POMDP solvers directly, we do expect that collaboration problems that are easier to solve by decision-theoretic approaches should also be easier for other action-selection mechanisms.

4.2 Collaborative Scenarios and Action Selection Algorithms

We used the Multi-Agent Decision Problem Toolbox, developed by Spaan and Oliehoek (2008) to examine the effects of coordination rules. This is an open-source toolbox that implements a variety of Dec-POMDP solvers and benchmark problems. We selected three different scenarios with three hand-authored coordination rules:

- Intersection: Two vehicles at an intersection with actions to move forward or wait. If both cars move forward at the same time, they crash resulting in a negative reward. If both cars make it through, there is a positive reward.
  - Rule: Car 1 will wait.
- Recycling Robots: Two robots either recycle a small amount, recycle a large amount, or recharge. While the robots can recycle small pieces independently, they must collaborate to recycle a large piece, which results in the largest reward.
  - Rule: Robot 1 will not attempt to recycle large amounts.
- Missile Strike: Two salvos of missiles are launched at high-priority and low-priority targets. The targets have countermeasures that may jam the communications or destroy the salvos, and the salvos can change what they are targeting.
  - Rule: The first salvo will engage the low-priority target.

For each of these scenarios, we generated policies for four planning horizons using the default optimal and approximate algorithms from the toolbox:

- Optimal: General multi-agent A* (Oliehoek et al., 2008)
- Approximate: Forward policy search with alternating maximization (Emery-Montemerlo et al., 2004).

These methods are deterministic and therefore we collect a single data point for each of 24 conditions (three scenarios $\times$ four time horizons $\times$ two algorithms).[3] To ensure our results are not due to the structure of particular Dec-POMDPs, we vary the scenario. By varying the time horizon, we create conditions requiring differing computational resources. By including optimal and approximate algorithms, we can measure the negative effects of the coordination rules on the optimal solutions, as well as identify any possible positive effects due to the combination of reduced search space and approximate methods. For each condition, we measured the amount of time required to generate a policy along with the expected utility of the policy, with and without the coordination rule. If solving the Dec-POMDP either with or without the coordination rule resulted in exceeding the resource bounds of the MADP-toolbox on a standard desktop, we report it as being unable to find a solution.

4.3 Results

Coordination rules provided an order-of-magnitude time savings with minimal effects on solution quality. In each scenario, the coordination rule let the solver find a solution for a longer horizon than the uncoordinated case. Furthermore, Figure 3 illustrates that on problems with the same planning horizon that were solved in a nontrivial amount of time, the same problems without coordination

3. This set of conditions is preliminary. Future studies should vary additional aspects of each condition to ensure measured differences are from the independent variables and not other aspects of the problem setup (e.g., initial conditions).

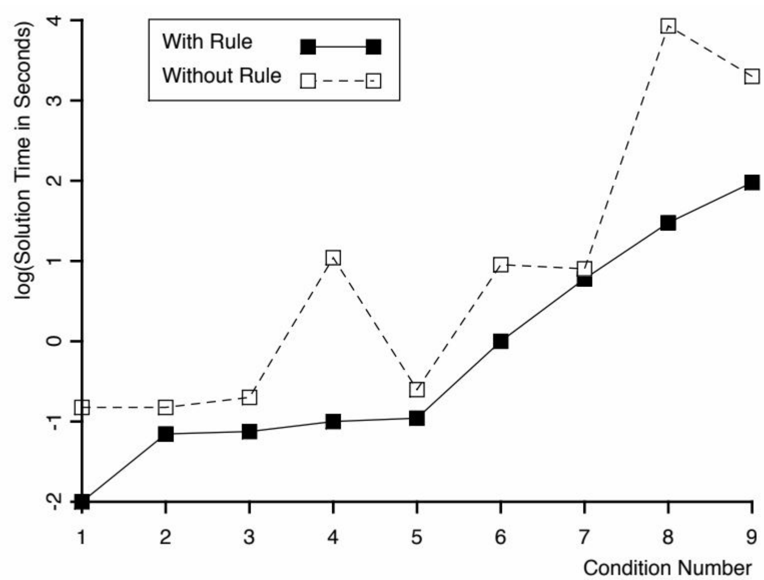

Figure 3. Comparison of planning times with and without coordination rules for trials in which solutions were found for both conditions ordered by the amount of time required to solve with the coordination rule.

rules took substantially longer to solve. In the most extreme example (the recycling robots domain with a planning horizon of four), the solution time with the coordination rule was 22 seconds, while without the rule, it took 86 minutes. These results support the claim that coordination rules enable faster planning for longer horizons.

The second issue concerns the effects of coordination rules on solution quality. For each planning horizon and scenario that generated policies with and without coordination rules, using coordination rules produced results that were, on average, three percent worse. In the worst case, the coordination rule resulted in a solution that was six percent worse for the longest planning horizon on the recycling robots. This is because it prevents the highest reward action from being taken. This result indicates that while coordination rules reduce the quality of the solution, they do not do so substantially.

The third issue concerns the possibility of the quality of approximate solutions improving with the use of coordination rules. In the missile strike scenario, the policies generated by the approximate solution method were one percent better with the coordination rule than those without. This result supports the claim that, when used with approximate solvers, coordination rules may result in better plan solutions by enabling the solver to explore a greater percentage of the policy space.

5. Related Work on Collaboration

Most cognitive systems research on collaboration focuses on beliefs and mutual beliefs of the agents. Bello (2012) demonstrates the importance of maintaining representations of others' beliefs to effectively reason about the world. Kennedy et al. (2008) emphasize the importance of learning models of others' capabilities when reasoning about what they might do. Lathrop et al. (2011) illustrate how spatial representations provide functional advantages for considering teammates' perspectives. Our focus is not on how others' beliefs are represented, but instead on how an understanding of possible futures can facilitate collaboration. An important extension to our approach would be to include explicit models of others' beliefs into the factored envisionment and explore the implications of such beliefs on uncertainty metrics.

In addition to cognitive systems, our work builds on research from the knowledge representation, planning, and agents communities. An early approach to support multi-agent collaboration involves specifying joint commitments and mutual belief (Cohen & Levesque, 1990). Joint commitments provide a mechanism for one agent to promise another to either attain a state of mutual belief that the commitment has been satisfied or to communicate that it is impossible. The key benefit of this framework is that it prevents agents from dropping out of collaborative behavior as soon as uncertainty arises, as they know their teammates will inform them if they are working toward impossible goals.

Building on this approach is the idea of SharedPlans (Grosz & Kraus, 1996), which were originally developed to further our understanding of dialog systems, but have been extended into a general model of collaborative behavior (Grosz & Kraus, 1999). SharedPlans evolve during their execution and maintain the mutual beliefs of the collaborating agents, along with important dependencies between their future actions.

Wooldridge and Jennings (1999) propose a definition of cooperative problem solving that goes beyond the focus of this work to include steps for recognizing collaborative problems and team formation. As in other work, they emphasize the importance of joint commitments, but they also introduce the concept of *conventions*, which are understandings within the team about when commitments may be abandoned. Our work complements these approaches in that, while they facilitate collaboration by agreeing on joint goals and subgoals *a priori*, coordination rules support collaboration by limiting the action space of the agents.

Although the above approaches use logic to support collaborative behavior, others in the multi-agent community view collaboration through mathematical frameworks describing series of decisions made by individual agents. These include decentralized partially observable Markov decision processes (Bernstein et al., 2000) and communicative multi-agent team decision processes (Pynadath & Tambe, 2002). Using a helicopter escort scenario, the latter authors show how different communication policies affects the expected utility of the teams plans. Our work contributes to these lines of inquiry in two ways: (1) we construct the state-action space through our plan-with-options domain representation and the envisionment process, and (2) we introduce a heuristic that reduces the state-action space by using predictability to limit individual actions. Further study of the costs and benefits of such heuristics would be an important direction for future work.

6. Conclusion

As cognitive systems become increasingly prevalent in the environment, our ability to collaborate with them, as well as their ability to collaborate with each other, will determine their utility to society. This paper introduced coordination rules as a way to reduce the space of possible actions for an autonomous agent that it conveys to teammates to facilitate interaction when communication is limited. We explored three issues around developing cognitive systems with coordination rules:

1. *How can one represent the space of possible futures in a multi-agent system?* We introduced plan-with-options, which provides a natural way to specify the space of coordinated behaviors without unnecessarily removing agent autonomy. We also demonstrated (on scenarios with increasing complexity) how interaction-based factored envisioning compactly represents the effects of multi-agent plan executions.
2. *How can one select which future actions to ignore if communication is cut?* We defined two classes of within-team uncertainty metrics, and we showed how they work in different settings with a case study of three scenarios.
3. *How does removing actions from consideration impact performance?* We demonstrated that coordination rules facilitate action selection in three scenarios by removing complexity from the state-action space. This accelerates planning times and allows for longer planning horizons while having only minor effects on plan quality.

Our highest priority for future work is to design and implement a cognitive system based on these findings that generates coordination rules and shares them with collaborators. After building this system, we will revisit the three claims and test them more thoroughly.

In the course of this project, our team has interacted with others on the design of open architectures for collaborative autonomous systems. Through these interactions, it has quickly become clear that different designers want different models of the domain and environment. In this paper, we presented a qualitative evaluation of the state-action space to represent possible futures, and we used Dec-POMDPs to capture the action selection problem for decision-theoretic planning. Ideally, these should be organized into a hierarchy or otherwise formally related. We believe there is a need for new theories of domain knowledge and its acquisition at multiple scales to use in designing agents that act as long-lived collaborative cognitive systems.

Acknowledgements

This research was developed with funding from the Defense Advanced Research Projects Agency. The views, opinions, and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. The authors also thank Pat Langley, Filip Dvorak, and reviewers for the Fourth Annual Conference on Advances in Cognitive Systems for their helpful comments and suggestions.

References

Alford, R., Kuter, U., & Nau, D. S. (2009). Translating HTNs to PDDL: A small amount of domain knowledge can go a long way. *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence* (pp. 1629–1634). Pasadena, CA: IJCAI.

Bello, P. (2012). Cognitive foundations for a computational theory of mindreading. *Advances in Cognitive Systems*, *1*, 59–72.

Bernstein, D. S., Zilberstein, S., & Immerman, N. (2000). The complexity of decentralized control of Markov decision processes. *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence* (pp. 32–37). Burlington, MA: Morgan Kaufmann.

Clancy, D. J., & Kuipers, B. (1997). Model decomposition and simulation: A component based qualitative simulation algorithm. *Proceedings of the Fourteenth National Conference on Artificial Intelligence* (pp. 118–124). Providence, RI: AAAI Press.

Cohen, P. R., & Levesque, H. J. (1990). Intention is choice with commitment. *Artificial Intelligence*, *42*, 213–261.

Do, M. B., & Kambhampati, S. (2003). SAPA: A multi-objective metric temporal planner. *Journal of Artificial Intelligence Research*, *20*, 155–194.

Emery-Montemerlo, R., Gordon, G., Schneider, J., & Thrun, S. (2004). Approximate solutions for partially observable stochastic games with common payoffs. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 136–143). New York: Association for Computing Machinery.

Erol, K., Hendler, J., & Nau, D. S. (1994). HTN planning: Complexity and expressivity. *Proceedings of the Twelfth National Conference on Artificial Intelligence* (pp. 1123–1128). Seattle, WA: AAAI Press.

Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated planning: Theory & practice*. San Francisco, CA: Elsevier.

Grosz, B. J., & Kraus, S. (1996). Collaborative plans for complex group action. *Artificial Intelligence*, *86*, 269–357.

Grosz, B. J., & Kraus, S. (1999). The evolution of SharedPlans. In M. J. Wooldridge & A. Rao (Eds.), *Foundations of rational agency*. Dordrecht, Netherlands: Springer.

Hinrichs, T. R., Forbus, K. D., de Kleer, J., Yoon, S., Jones, E., Hyland, R., & Wilson, J. (2011). Hybrid qualitative simulation of military operations. *Proceedings of the Twenty-Third Innovative Applications of Artificial Intelligence Conference* (pp. 1655–1661). San Francisco: AAAI Press.

Kennedy, W. G., Bugajska, M. D., Adams, W., Schultz, A. C., & Trafton, J. G. (2008). Incorporating mental simulation for a more effective robotic teammate. *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence* (pp. 1300–1305). Chicago, IL: AAAI Press.

Kuipers, B. (1994). *Qualitative reasoning: Modeling and simulation with incomplete knowledge*. Cambridge, MA: The MIT Press.

Langley, P. (2012). The cognitive systems paradigm. *Advances in Cognitive Systems*, *1*, 3–13.

Lathrop, S. D., Wintermute, S., & Laird, J. E. (2011). Exploring the functional advantages of spatial and visual cognition from an architectural perspective. *Topics in Cognitive Science*, *3*, 796–818.

McDermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A., Veloso, M., Weld, D., & Wilkins, D. (1998). *PDDL - The planning domain definition language* (Technical Report TR-98-003). Center for Computational Vision and Control, Departments of Electrical Engineering and Computer Science, Yale University, New Haven, CT.

Nau, D. S., Au, T.-C., Ilghami, O., Kuter, U., Murdock, J. W., Wu, D., & Yaman, F. (2003). SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research*, *20*, 379–404.

Oliehoek, F. A., Spaan, M. T., & Vlassis, N. A. (2008). Optimal and approximate q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, *32*, 289–353.

Pynadath, D. V., & Tambe, M. (2002). The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of Artificial Intelligence Research*, *16*, 389–423.

Shannon, C. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*, 379–423, 623–656.

Spaan, M. T., & Oliehoek, F. A. (2008). The MultiAgent Decision Process toolbox: Software for decision-theoretic planning in multiagent systems. *Proceedings of the AAMAS Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains* (pp. 107–121). Estoril, Portugal: IFAAMAS.

Weld, D., & de Kleer, J. (Eds.). (1989). *Readings in qualitative reasoning about physical systems*. San Francisco, CA: Morgan Kaufmann.

Wooldridge, M., & Jennings, N. R. (1999). The cooperative problem-solving process. *Journal of Logic and Computation*, *9*, 563–592.

Younes, H. L. S., & Littman, M. L. (2004). *PPDDL1.0: An extension to PDDL for expressing planning domains with probabilistic effects* (Technical Report CMU-CS-04-167). School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.