

Decentralized Models for Use in a Real-World Personal Assistant Agent Scenario

Christopher Amato and Nathan Schurr and Paul Picciano

Aptima, Inc.

12 Gill Street, Suite 1400

Woburn, MA 01801

{camato,nschurr,ppicciano}@aptima.com

Abstract

Many approaches have been introduced for representing and solving multiagent coordination problems. Unfortunately, these methods make assumptions that limit their usefulness when combined with human operators and real-life hardware and software. In this paper, we discuss the problem of using agents in conjunction with human operators to improve coordination as well as possible models that could be used in these problems. Our approach – Space Collaboration via an Agent Network (SCAN) – enables proxy agents to represent each of the stakeholder agencies in a space setting and shows how the SCAN agent network could facilitate collaboration by identifying opportunities and methods of collaboration. We discuss this approach as well as the challenges in extending models to 1) take advantage of human input, 2) deal with the limited and uncertain information that will be present and 3) combat the scalability issues in solution methods for a large number of decentralized agents. As a first step toward providing rich models for these domains, we describe a method to bound the solution quality due to bounded model uncertainty.

Introduction

In today's world, people are increasingly connected to many other people and many sources of information, providing more opportunities than ever for tasks such as working more efficiently with others and discovering helpful information. Unfortunately, the vast scale of information can be overwhelming. As computers become more powerful and pervasive, software in the form of personal assistant agents can provide support in many situations.

For example, with the increased deployment and use of space assets, a number of interesting and challenging problems have come to the fore. The persistent nature of space surveillance (i.e., 24/7 operations), the mass of data, the varied data sources, and the heterogeneous needs of consumers and producers throughout the community all point to a pressing need for an enhanced Space Situation Awareness (SSA) picture, one that can only be achieved by a coordinated, collaborative approach to the collection, dissemination and sharing of information.

For the foreseeable future, the challenges in the U.S. National Space Policy will demand human-in-the-loop participation. This is particularly relevant given the amount and

importance of data and knowledge not held in any database, or streaming in bits through the ionosphere, but that resides in the minds of individuals or the institutional knowledge of a team. The criticality of an enhanced situation awareness and its subsequent deployment suggests a focus on three distinct areas:

- Compile an index of human knowledge so that both the requestors and receivers of information know what data are available, who can support timely anomaly prediction and resolution, and how to determine the best courses of action going forward.
- Establish common methods and procedures for collaboration, so that communication between and among interested and relevant parties can be efficiently facilitated, while preserving the security of the information being exchanged.
- Integrate these methods within the existing workflow, obviating the need to add additional tools.

While these three foci are distinct, their solution requires a coherent, integrated approach, viewing the space domain as a socio-technical system, within which the human plays an integral part. Several diverse agencies are stakeholders and produce and/or consume information related to space. Each agency has its own organizational structure and protocols, so the solution must be versatile enough to allow for effective inter-agency collaboration while still maintaining the standard practices of each.

Further, the human operator network alone is not sufficient to create adequate SSA. There is an enormous amount of data recorded from sensors on satellites, ground stations and other periodic collectors. The human operator is overloaded by the transmission, capture, cataloging, and sense making of these data. The optimal solution for enhanced SSA is a synergistic, collaborative effort between networks of humans and automated agents.

The Space Collaboration via an Agent Network (SCAN) approach enables proxy agents to represent each of the stakeholder agencies for space and to facilitate collaboration by identifying opportunities and methods for collaboration. This allows humans to have access to large amounts of data through their agents, while personalizing them given the users specific preferences and capabilities. These agents can then communicate with each other with varying levels

of input from the user for tasks ranging from retrieving information, securing the services of another agency or team formation for complex tasks.

Many agent models have been developed for solving decentralized problems requiring cooperation. For SCAN, we assume a sequential problem in which series of decisions need to be made, each affecting the subsequent. Limiting the sequential models for multiagent cooperation to those based on decision theory, a large number of models can still be used (Becker, Lesser, and Zilberstein 2004; Bernstein et al. 2002; Boutilier 1999; Nair et al. 2005; Guestrin, Koller, and Parr 2001). Nevertheless, each of these models makes assumptions that cause it to be inappropriate for a domain such as SCAN. These assumptions include: a full model of the problem with the value other agents derive from all actions or the results of these actions, perfect knowledge of the current state of the system (or a common estimate), a shared objective function, a centralized solution method. An ideal model for our domain would relax these assumptions to permit only local knowledge and solution methods, self-interested agents, model uncertainty and independence between groups of agents.

The remainder of this paper is organized as follows. We first discuss the SCAN domain in more detail as well as the progress made to date. We also discuss several models that could be used to represent this problem and their shortcomings. Finally, we discuss key assumptions that we are interested in relaxing in these approaches and the challenges involved in doing so. The goal in this project is to produce a system of personal assistant agents with accompanying solution methods that maintain tractability and high solution quality.

Overview of SCAN approach

The approach used in SCAN represents each of the stakeholder agencies for space with proxy agents and facilitates collaboration by identifying opportunities and methods for collaboration. It demonstrates the suitability of our human-agent solution to address the three critical aspects above, that is: 1) Knowledge and mental models of human operators, 2) Collaboration methods and barriers to collaboration and 3) Integrating solutions that conform to preferred workflows.

A variety of factors shape the opportunities for collaboration and the constraints on collaboration. The SCAN Agent architecture incorporates these components into the SCAN Proxy Agent Architecture. The need to collaborate varies based on an interaction of the demands of the mission, the architecture of the team and its resources, and the distribution of expertise within the team. Collaboration becomes necessary when mission-critical expertise is distributed among multiple people, and resources and responsibilities are likewise divided between people.

A team's ability to collaborate depends on a number of factors, including available technology, team members collaboration skills, and team composition. The factors affecting the ability to collaborate directly affect the products of collaboration (e.g., assessments and plans), as well as the apparent coordination of the team as a whole. Collaborative critical thinking is intended to provide active support to

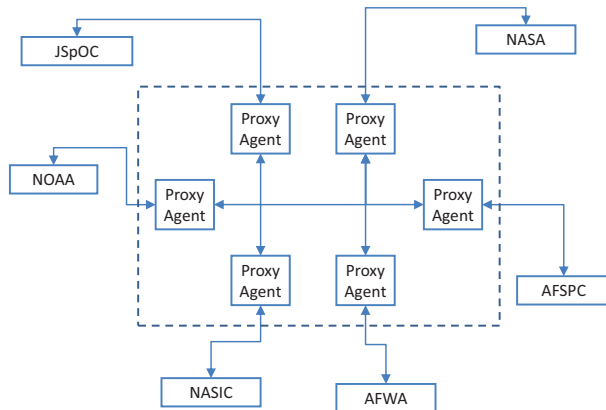


Figure 1: SCAN Proxy Agents being used to facilitate collaboration between the heterogeneous stakeholders in space.

teams above and beyond the three factors affecting a team's ability to collaborate. Figure 1 shows a top-level diagram of our proposed solution: Space Collaboration via an Agent Network (SCAN). It shows proxy agents, each of which represents a stakeholder for space. We implemented an initial version of this infrastructure by developing agent interactions to carry out the use case(s), and tested the resultant model by generating results to show dynamic constraints, changing parameters, and making modifications to the use cases(s) that show the benefit of agent use. This solution will fit integrally into the current workflow, without the need for additional tools for users to learn, and will make privacy issues explicit.

It is important to note that our approach to SCAN is to have the human remain in control and drive the collaboration. The proxy agent is designed to facilitate the human operators and point them in the right direction to improve interagency collaboration. Human operators will not be surprised or superseded by the SCAN proxy agent's actions.

The SCAN framework

In conceiving an agent-based architecture, we identified the following benefits to individual agencies of agent-based collaboration:

1. Fusion/Integration of agency plans
2. Locating potential collaboration opportunities
3. Forewarning: An agency may anticipate future events based on an alert from another agency
4. Increased Quality/Accuracy of data
5. Facilitating levels of automated collaboration
6. Detecting collaboration failures
7. Adjusting to realtime changes in policy/capabilities
8. Learning preferences of agencies
9. Persuading agencies to cooperate through incentives

We selected areas (1), (2), and (7) as areas for further study.

Use case

In order to study the above, we constructed a working Use Case. The requirements for the Use Case were:

Scalability: The initial Use Case should describe a simple problem that can easily scale to more complex problems. This allows us to demonstrate our solution initially in an easily controllable environment, but also set the stage for our solution to handle a larger set of more complex problems.

Generalizability: The Use Case should be representative of a general class of collaboration problems, so that the solution will be applicable to the needs of other agencies as well as those mentioned in the Use Case. The Use Case was based on interviews with space weather analysts from the Air Force Weather Agency (AFWA), and their information needs as primary producers of information on environmental effects to agencies such as NASIC, JSpOC, and the Air Operations Center (AOC)

We identified two important facets to examine in developing the use case: (P1) How does the SCAN agent know when to initiate collaboration? (P2) What is the message structure necessary to handle collaboration? The message structure will be driven by the type of collaboration, therefore (P1) must be considered before (P2). For (P1), the following sequence of steps should take place in the Use Case: (S1) NASIC's communication to the satellite goes down (S2) NASICs SCAN agent attempts to diagnose the reasons (S3) Two causes are identified, a solar flare or a system problem (S4) Collaboration IDs are generated for each of the two causes (S5) NASICs SCAN agent sends a query to the other agents to collaborate (S6) Agents on behalf of the other agencies provide an affirmative response to the collaboration request (S7) The agencies are connected.

This protocol is motivated by Signaling System 7 (SS7) of the telephone network (Modarressi and Skoog 1990) which separates data from service. This property is particularly desirable, as it allows an agency to limit the amount of data it discloses to non-collaborating agencies. In the future, the topology of the SCAN agents will be further refined.

Basic model

From the perspective of a SCAN agent receiving collaboration requests, this agent is called the Collaboration Facilitator (CF). The requesting agent is called the CR. As stated previously, each SCAN agent will have a model of the other agents and agencies. The model will be used to derive such information as an arrival rate of resource requests, $\lambda(i)$. Based on the model of the requests coming in, the model will also predict a service rate of these requests, $\mu(i)$. The model will also predict the benefits of each collaboration from each organization. Likewise, from the perspective of the CF, it is possible that a collaboration initiated by another agency will require collaboration with a third agency, and that only the CF and not the CR knows about this collaboration. In this case, it will NOT be the role of the CF to request collaboration with the third agency. This could reproduce the well-known "dining philosophers" problem from Operating

Systems literature which is essentially that a bottleneck results when multiple processes compete for a limited set of resources. As a consequence of that bottleneck, system performance is degraded. Instead, the CF agent passes the information about the 3rd party back to the CR agent and the CR agent submits the request for collaboration to the third agency.

The motivation behind this model is that a SCAN agent receiving a collaboration request may be able to make decisions like: (1) "Well, technically I could collaborate you; after all I have enough resources to do it. However I know I'm about to get some high priority requests from important agencies soon, so... collaboration request denied." (2) "I can't collaborate with you right now. But my model says I'll be free in a few hours, do you want to book my time in advance?"

Multiagent models of interest

To model the type of problems described above, we need to consider the following requirements: sequential decisions, uncertainty regarding outcomes and other agents' decisions and decentralized control. Sequential decisions are needed because agents' decisions at the current time will affect the future outcomes. That is, suggesting to collaborate with one agency rather than another may have consequences in terms of time and resources required, affecting future collaboration possibilities. Uncertainty is present in these scenarios as well considering that information may be unavailable or outside forces could cause the situation to change. Also, users may decline collaboration and the actions of the other agents are often unseen, causing uncertainty about the other agents. The system must also be decentralized for similar reasons. Agents must make choices based solely on local information due to the lack of updates from other agents. This local information may also be uncertain or incomplete information about the human user or other humans and agents in the system.

This type of modeling can be accomplished with decision-theoretic approaches. In general, these representations have probabilities which represent the uncertainty in action outcomes and seek to maximize an objective function in an attempt to optimize the sequence of decisions by the agents. As agents are built for ever more complex environments, methods that consider the uncertainty in the system have strong advantages. Developing effective frameworks for reasoning under uncertainty is a thriving research area in artificial intelligence and we discuss some of these approaches below.

DEC-POMDPs

A DEC-POMDP (Bernstein et al. 2002) can be defined with the tuple: $\langle I, S, \{A_i\}, P, R, \{\Omega_i\}, O, T \rangle$ with

- I , a finite set of agents
- S , a finite set of states with designated initial state distribution b_0
- A_i , a finite set of actions for each agent, i
- P , a set of state transition probabilities: $P(s'|s, \vec{a})$, the probability of transitioning from state s to s' when the set of actions \vec{a} are taken by the agents

- R , a reward function: $R(s, \vec{a})$, the immediate reward for being in state s and taking the set of actions \vec{a}
- Ω_i , a finite set of observations for each agent, i
- O , a set of observation probabilities: $O(\vec{o}|s', \vec{a})$, the probability of seeing the set of observations \vec{o} given the set of actions \vec{a} was taken which results in state s'
- T , a horizon or number of steps after which the problem terminates

A DEC-POMDP involves multiple agents that operate under uncertainty based on different streams of observations. At each step, every agent chooses an action based on their local observation histories, resulting in a global immediate reward and a local observation for each agent. Note that because the state is not directly observed, it may be beneficial for the agent to remember its observation history. A *local policy* for an agent is a mapping from local observation histories to actions while a *joint policy* is a set of local policies, one for each agent in the problem. The goal is to maximize the total cumulative reward until the horizon is reached, beginning at some initial distribution over states. In the infinite-horizon problem, T is infinity and the decision making process unfolds over an infinite sequence of steps. In order to maintain a finite sum over the infinite-horizon, in these cases a discount factor, $0 \leq \gamma < 1$, is employed.

The DEC-POMDP model is very general, but has a very high complexity (NEXP-complete¹). Algorithms for solving DEC-POMDPs also typically assume that all model parameters are known in advance and the solution is found in a centralized manner, providing policies that each agent can implement in a decentralized way. This model also assumes no communication (except as part of the observations) and full cooperation between the agents. The high complexity and large number of assumptions make the DEC-POMDP model currently inappropriate for the scenarios we are interested in for SCAN with a large number of agents with limited knowledge of the full problem model.

Communication and learning have also been studied in the DEC-POMDP model. Recent work has examined using communication for online planning in DEC-POMDPs (Wu, Zilberstein, and Chen 2009). This type of work could be useful in our context because communication is used when inconsistencies are detected between the agents. Unfortunately, it still requires knowledge of the full model and a DEC-POMDP to be solved for some number of steps. Rather than assuming the model is known, it can also be learned or policies can be learned directly (Dutech, Buffet, and Charpillet 2001; Peshkin et al. 2000). Direct policy learning may be an appropriate approach in that it does not require a model, but in order to calculate the gradient the model must be sampled. This sampling requires many instances and still is not guaranteed to converge to an equilibrium.

DEC-MDPs and MMDPs

We can restrict DEC-POMDPs in various ways to simplify solution calculations. For instance, we can assume that each

¹This results in doubly exponential complexity as long as $P \neq NP$.

agent can fully observe its own local state, but not the global state. This is a form of the DEC-MDP model (Becker, Lesser, and Zilberstein 2004), which has been shown to have more efficient solutions in some cases, but in general has the same complexity as the general model (NEXP-complete) (Bernstein et al. 2002). If we consider problems in which the agents have independent transitions and observations and a structured reward model (IT-IO DEC-MDPs), the complexity drops to NP-complete (Becker et al. 2004). Recent work has shown that subclasses of DEC-POMDPs which have independent rewards, but dependent observations and transitions as well as those with certain structured actions retain the same complexity as the general problem (Allen and Zilberstein 2009). Some other modeling assumptions and the resulting complexity are studied in (Goldman and Zilberstein 2004), but none of these seems appropriate for our case as they continue to assume centralized solution methods and knowledge of the model.

DEC-MDPs can be further simplified by assuming that the state of the problem is fully observed by each agent, resulting in a multiagent MDP (MMDP) (Boutilier 1999). There has been work on efficient planning solutions for these problems (modeled as factored MDPs) (Guestrin, Koller, and Parr 2001), which allow algorithms to exploit independence between agents. While this is a useful quality (which is discussed in more detail in the context of ND-POMDPs), centralized solution methods are used and full model knowledge is required, limiting the applicability of these models to our scenario.

ND-POMDPs

Another way of simplifying DEC-POMDPs is to assume agents interact in a limited way based on locality. This assumption often considers agents that can only interact with their neighbors and thus is more scalable in the number of agents in the problem. This has been studied using the networked distributed POMDP (ND-POMDP) model for the finite-horizon case (Nair et al. 2005) as well as a more general factored models (Oliehoek et al. 2008; Witwicki and Durfee 2010).

The ND-POMDP model is similar to the DEC-POMDP model except for the following: states can be factored $S = \times S_i \times S_u$ for each set of states for each agent i and an unaffected state set, actions and observations can be factored similarly with $A = \times A_i$ and $\Omega = \times \Omega_i$ for each agent, transition independence where the unaffected state and each agent transition independently $P(s'|s, a) = P(s'_u|s_u) \prod_i P(s'_i|s_i, a_i)$ and observation independence where $O(o|s', a) = \prod_i O_i(o'_i|s_i, a_i)$. Also, rewards can be broken up based on neighboring agents and summed as follows:

$$R(s, a) = \sum_l (s_l, \dots, s_{lk}, s_u, \langle a_l, \dots, a_{lk} \rangle)$$

where l represents a group of $k = |l|$ neighboring agents. The ND-POMDP model also assumes that the belief state (the state distribution estimate) is known by all agents.

A solution to an ND-POMDP can be computed in a distributed manner using message passing to converge to a local

Model	Pros	Cons
DEC-POMDPs	Very rich model of cooperative agents	Assumes known model parameters, fully cooperative, high complexity
IO-IT DEC-MDPs	Somewhat rich, less complex than full model	Limiting model assumptions (agents interact only through rewards), Assumes known model parameters, fully cooperative
MMDPs	Low complexity	Limiting model assumptions (centralized knowledge), Assumes known model parameters, fully cooperative
ND-POMDPs	Exploits locality, distributed solution, less complex than full model	Limiting model assumptions (very limited interaction), Assumes some known model parameters, fully cooperative
Graphical games	Allows self interest, exploits locality	Assumes some known model parameters, not sequential
I-POMDPs	Allows self interest, exploits locality	Assumes known model parameters, high complexity

Table 1: Model pros and cons for distributed personal assistant domains.

optimum. This approach can be very efficient in cases where the agents are only loosely connected. Nevertheless, strong assumptions are still made such as knowing the model in the local neighborhood and additive rewards across neighborhoods.

Game theoretic representations

If we assume that agents may have different interests that may not necessarily align perfectly (which is likely to be the case in many real-world scenarios), we could use a game theoretic approach. A model that retains the idea of agents loosely connect into neighborhoods is the graphical game (Kearns, Littman, and Singh 2001). This approach combats the tabular (and thus exponential) representation of n -agent games by using an undirected graph where two agents are connected if they can affect each others' payoffs. Assumptions in this model are that each agent can calculate a best response, which means that it knows the payoffs for the different actions given the other agents' choices. This is also not a sequential model, so all possible horizon T policies would have to be considered as actions, resulting in an exponential increase in action size and intractability to solve.

Sequential game theoretic models have also been studied. A generalization of DEC-POMDPs to the case where agents may have different rewards results in a partially observable stochastic game (POSG). Thus, if we assume POSGs with common payoffs for all agents, it is equivalent to the DEC-POMDP model (Hansen, Bernstein, and Zilberstein 2004). This model relaxes the assumption that all agents are fully cooperative, but retains the intractability and knowledge of the model assumed by the DEC-POMDP. Another game theoretic view of multiagent POMDPs from the Bayesian perspective is the interactive POMDP (I-POMDP) model (Gmytrasiewicz and Doshi 2005). An extension of the I-POMDP model to graphical representations, allowing agent independence to be exploited is the interactive dynamic influence diagram (I-DIDs) (Doshi, Zeng, and Chen 2009). While both of these models are interesting in that they can represent problems with estimates of other agent behavior based on observations that are seen, a full model is again assumed and complexity is at least as high as the corresponding DEC-POMDP models.

Extensions

As shown in Table 1, the models discussed have varying pros and cons. It is often the case that a choice must be made which balances the trade-off between model richness and computational tractability. Incorporating richness into models to allow them to more accurately represent real-world problems as well as developing models with a lower computational footprint, while still preserving the structure of the problem are both very important goals. We plan to pursue extensions to these models which fit the personal assistant agent domain, without markedly increasing complexity. We discuss one such step below.

Bounded model uncertainty

As a first step towards relaxing assumptions of these models, we consider the case where there is model uncertainty. We begin with the simple case where a model is known, but a parameter may only be known within some error ϵ . That is, in the reward case for DEC-POMDPs, reward error could cause the true reward to be in the interval $[R(s, \vec{a}) - \epsilon, R(s, \vec{a}) + \epsilon]$. Similar model uncertainty can cause the transition and observation model parameters to only be known within ϵ . We consider the DEC-POMDP model, but similar results can be shown for the other models.

We first show that the value of a set of policies can be bounded when such an ϵ error is introduced.

Assuming policy \vec{p} for the set of agents, we can evaluate this policy with

$$V(s, \vec{p}) = R(s, \vec{a}_{\vec{p}}) + \gamma \sum_{s', o} P(s'|s, \vec{a}_{\vec{p}})P(\vec{o}|s', \vec{a}_{\vec{p}})V(s', \vec{p}_{\vec{o}})$$

where $\vec{a}_{\vec{p}}$ represents the first action defined by \vec{p} and $\vec{p}_{\vec{o}}$ represents the next step policy that is followed after \vec{o} is seen by the agents.

We can define $V_{max}(s, \vec{p})$ as the maximum reward found by introducing a maximum error in the reward function as follows $V_{max}(s, \vec{p}) =$

$$R(s, \vec{a}_{\vec{p}}) + \epsilon + \gamma \sum_{s', o} P(s'|s, \vec{a}_{\vec{p}})P(\vec{o}|s', \vec{a}_{\vec{p}})V_{max}(s', \vec{p}_{\vec{o}})$$

Because this is an additive error, we get the following

$$\begin{aligned} &= R(s, \vec{a}_{\vec{p}}) + \epsilon + \gamma \sum_{s', o} P(s'|s, \vec{a}_{\vec{p}}) P(o|s', \vec{a}_{\vec{p}}) [V(s', \vec{p}_{\vec{o}}) + V_{\epsilon}] \\ &= R(s, \vec{a}_{\vec{p}}) + \epsilon + \gamma \sum_{s', o} P(s'|s, \vec{a}_{\vec{p}}) P(o|s', \vec{a}_{\vec{p}}) V(s', \vec{p}_{\vec{o}}) + \gamma V_{\epsilon} \end{aligned}$$

This is true because V_{ϵ} does not depend on s or \vec{o} (observe that on the last level of recursion $\sum_{s', o} P(s'|s, \vec{a}) P(o|s', \vec{a}) \epsilon = \epsilon$). Thus, it can be seen that $V_{\epsilon} = \epsilon + \gamma \epsilon + \dots + \gamma^{T-2} \epsilon$ and

$$V_{max}(s, \vec{p}) = V(s, \vec{p}) + \frac{1 - \gamma^{T+1}}{1 - \gamma} \epsilon$$

A similar result can be found for the minimum value given the model parameters. Therefore, the true value for any state will be will be

$$V_{true} = \left[V(s, \vec{p}) - \frac{1 - \gamma^{T+1}}{1 - \gamma} \epsilon, V(s, \vec{p}) + \frac{1 - \gamma^{T+1}}{1 - \gamma} \epsilon \right]$$

This analysis can also be conducted for both the transition and observation models. The result of an ϵ error in both models is $\frac{1 - \gamma^T}{1 - \gamma} \gamma |S| |\Omega| \epsilon^2$ or $\frac{1 - \gamma^T}{1 - \gamma} \gamma |S| \epsilon$ for just the transition model and $\frac{1 - \gamma^T}{1 - \gamma} \gamma |\Omega| \epsilon$ for just the observation model. When the horizon is infinite, $1/1 - \gamma$ will replace $1 - \gamma^{T+1}/1 - \gamma$.

While this type of model uncertainty can be bounded, it does not mean that all current solution techniques may be used without change. For instance, pruning in dynamic programming for DEC-POMDPs (Hansen, Bernstein, and Zilberstein 2004; Bernstein et al. 2009) must be adapted to remove only policies that are dominated by 2ϵ now to ensure they truly have lower value. This will result in more policies being retained and thus higher resource usage. Similarly, changes must be made in other algorithms for other models as well when an ϵ error is introduced.

Conclusion

In this paper, we discussed a real-world domain for facilitating collaboration between organizations and people, the SCAN proxy agents. We described the characteristics of this domain which require sequential and decentralized decision-making. Various decision-theoretic models for representing these problems are presented along with their shortcomings in this domain. In order to begin to address these shortcomings, we present one approach for bounding the value error when the model error can be bounded by ϵ . In the future, we are interested in further extending decision-theoretic models so they can be applied in this context. This work will include topics such as fully decentralized decision-making, other forms of model uncertainty and self-interested behavior. These algorithms will be implemented and tested in the SCAN domain.

References

Allen, M., and Zilberstein, S. 2009. Complexity of decentralized control: Special cases. In Bengio, Y.; Schuurmans, D.; Lafferty, J.; Williams, C. K. I.; and Culotta, A., eds., *Advances in Neural Information Processing Systems*, 22. 19–27.

Becker, R.; Zilberstein, S.; Lesser, V.; and Goldman, C. V. 2004. Solving transition-independent decentralized Markov decision processes. *Journal of AI Research* 22:423–455.

Becker, R.; Lesser, V.; and Zilberstein, S. 2004. Decentralized Markov Decision Processes with Event-Driven Interactions. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, 302–309.

Bernstein, D. S.; Givan, R.; Immerman, N.; and Zilberstein, S. 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research* 27(4):819–840.

Bernstein, D. S.; Amato, C.; Hansen, E. A.; and Zilberstein, S. 2009. Policy iteration for decentralized control of Markov decision processes. *Journal of AI Research* 34:89–132.

Boutilier, C. 1999. Sequential optimality and coordination in multiagent systems. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 478–485.

Doshi, P.; Zeng, Y.; and Chen, Q. 2009. Graphical models for interactive POMDPs: Representations and solutions. *Journal of Autonomous Agents and Multi-Agent Systems* 18.

Dutech, A.; Buffet, O.; and Charpillet, F. 2001. Multi-agent systems by incremental gradient reinforcement learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 833–838.

Gmytrasiewicz, P. J., and Doshi, P. 2005. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research* 24:24–49.

Goldman, C. V., and Zilberstein, S. 2004. Decentralized control of cooperative systems: Categorization and complexity analysis. *Journal of Artificial Intelligence Research* 22:143–174.

Guestrin, C.; Koller, D.; and Parr, R. 2001. Multiagent planning with factored MDPs. In *Advances in Neural Information Processing Systems*, 15. 1523–1530.

Hansen, E. A.; Bernstein, D. S.; and Zilberstein, S. 2004. Dynamic programming for partially observable stochastic games. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, 709–715.

Kearns, M.; Littman, M. L.; and Singh, S. 2001. Graphical models for game theory. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*.

Modarressi, A., and Skoog, R. 1990. Signalling system no. 7: A tutorial. *IEEE Communications Magazine* July.

Nair, R.; Varakantham, P.; Tambe, M.; and Yokoo, M. 2005. Networked distributed POMDPs: a synthesis of distributed constraint optimization and POMDPs. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*.

Oliehoek, F. A.; Spaan, M. T. J.; Whiteson, S.; and Vlassis, N. 2008. Exploiting locality of interaction in factored Dec-POMDPs. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems*.

Peshkin, L.; Kim, K.-E.; Meuleau, N.; and Kaelbling, L. P. 2000. Learning to cooperate via policy search. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, 489–496.

Witwicki, S. J., and Durfee, E. H. 2010. Influence-based policy abstraction for weakly-coupled Dec-POMDPs. In *Proceedings of the Twentieth International Conference on Automated Planning and Scheduling*.

Wu, F.; Zilberstein, S.; and Chen, X. 2009. Multi-agent online planning with communication. In *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling*.