Selectively Injecting Organizational Influences into Decision-Theoretic Agents

Jason Sleight[1] and Edmund H. Durfee[1]

University of Michigan, Ann Arbor MI 48109, USA

Abstract. In this paper, we summarize our prior work on exerting organizational influence on a decision-theoretic agent by replacing components of its Dec-POMDP model. We show how replacing various components can correspond to different prior notions of organizational control, and summarize the impact that different replacements can have on the quality and overhead of coordinated behavior. The innovation of this paper is then presented, which is a principled methodology for organizational influence to be modeled as a partial, factored specification of a Dec-POMDP component, and for overlaying this onto (rather than simply replacing) an agent's pre-existing local component specification. By doing so, organizational influences can better complement the agents' local expertise. We illustrate these ideas and discuss some of the qualitative properties of our proposed technique through analytical evaluation across several performance dimensions.

1 Introduction

Organizational structuring is a widely adopted and often powerful tool for coordinating large groups of people to achieve common goals effectively and efficiently, by giving each person guidance in how to make local decisions that are useful to the collective endeavor. Multiagent systems research has investigated how organizational concepts and strategies can be modeled and utilized by computational agents, showing that organizations can increase the expected performance of large-scale, cooperative multiagent systems [1, 2]. Research also suggests that organizational control becomes increasingly effective as the number of agents increases, the time horizon increases, the system complexity increases, the system resources decrease, and/or the performance goals increase [3]. That these issues arise in realistic application domains has driven research into how to encode pertinent organizational influence for computational agents, and how to augment agent architectures to follow such influence.

In our previous work [4], we departed from the traditional view of what an organization is or could be, computationally, by beginning with a model of agent decision making based on decentralized partially observable Markov decision processes, Dec-POMDPs. Within this formal, well-defined decision framework, we then explored how various types of organizational control and influences can be captured in the different components of the framework, such as transition and reward functions, and empirically demonstrated the effectiveness of exercising organizational influence both within each of the components individually as well as across multiple components collectively.

In this paper, we extend that work to provide more flexible organizational specifications. Namely, the organizational influences should be able to complement, rather than override, the agents' pre-existing local skills, knowledge, and expertise captured in the components of its local decision-theoretic model. As a simple example, an agent may have carefully tuned parameters of its reward function based on its experiences. Organizational influence should be able to augment that reward function (e.g., to help an agent prioritize tasks with respect to its organizational role relative to other agents) without wiping out the expertise reflected in the tuned parameters. Our general philosophy is that an organizational design should exert influences based on a broader view of agents' social relationships, but defer to an agent's expertise for solving the local aspects of its problems.

To achieve this goal, we present a factored framework for both the agents' local models as well as the organizational specification, and provide a methodology for each agent to overlay its own organizationally-specified factors on top of its local model, effectively "masking" only those factors that the organization chooses to override. This allows each agent to continue to exploit its local knowledge, while still allowing for the organization to guide the agents into more globally desirable local decisions.

The rest of this paper is organized as follows. In Section 2 we provide a detailed description of the grid world firefighting domain, which we use for illustration in the remainder of the paper. We then discuss our previous results in injecting organizational influence by selectively replacing Dec-POMDP components of decision-theoretic agents (Section 3). Then, in Section 4 we extend that work by presenting our factored framework for the local models and organizational specification, and explain how each agent can overlay the organizationally specified factors on top of its pre-existing model. We then present an initial evaluation of the effectiveness of our factored framework across several performance dimensions in Section 5. Finally, we compare our research to related work and conclude (Section 6) with a summary of our presented work and our ongoing efforts.

2 Problem Representation

We adopt a standard Dec-POMDP decision model [5], $\mathcal{M} = \langle \mathcal{N}, S, \alpha, A, R, P, \Omega, O, T \rangle$, where: $\mathcal{N}$ is the set of n cooperative agents; S is the (finite) set of global states; α is a probability distribution over initial global states; A is the (finite) set of possible joint actions; R is the joint reward function; P is the joint transition function; Ω is the (finite) set of possible joint observations; O is the joint observation function; and T is the finite time horizon. Given a full specification of the Dec-POMDP, an optimal joint policy, π^*, can be formulated in principle. In practice, however, finding such a policy for anything but very simple problems (with few agents and small state and action spaces) is intractable

[5], and even if found, executing such a policy is problematic because it generally assumes that all agents have the same beliefs about the global state.

For these reasons, multiagent approaches to solving such problems often assume that each agent possesses a local view of the joint problem. As is customary in that work, we assume that state is factored: every state is represented using the same set of m_S state features, such that $S = F_1 \times \cdots \times F_{m_S}$, where F_j is the finite set of possible values for state feature j. Each agent i has a local state representation S_i consisting of a subset of the m_S features. We further adopt the common simplifying assumption of local full observability (each agent i can exactly observe the values of all of its local state's features)[1]. Given these assumptions, the local decision model $\mathcal{M}_i$ of an agent i represents a local MDP, $\mathcal{M}_i = \langle S_i, \alpha_i, A_i, R_i, P_i, T_i \rangle$, where local rewards, transitions, actions, etc. are defined over the states in S_i. Each agent can use its local MDP to compute its (optimal) local policy π_i^* with respect to $\mathcal{M}_i$. The joint policy is then simply defined as $\pi = \langle \pi_1^*, \pi_2^*, ..., \pi_n^* \rangle$.

To illustrate a problem of this type, we use a simplified firefighting scenario, where firefighting agents and fires to be fought exist in a grid world (Figure 1). The global state consists of the locations of the agents and the locations and intensities of the fires. Figure 1 shows an initial global state, where the locations of agents A1 and A2 are shown, along with positions of each fire Fx, where x is the current intensity of the fire in that position. Additionally, there are varying degrees of delay in each cell, $\delta_c \in [0, 1]$, which prevents movement into that cell in proportion to the degree of delay. In Figure 1, (H)igh, (M)edium, and (L)ow delay correspond to δ equal $0.8, 0.5,$ and 0.0 respectively. The value of the delay in each cell is fully observable to each agent. Each agent has 6 actions: a NOOP action that makes no change to the world state; 4 possible movement actions (N, S, E, W) that move the agent one cell in the specified direction with probability $1 - \delta_{c\_dest}$ (and equates to a NOOP if there is no cell in that direction or delay prevents it from moving); and a fight-fire (FF) action that decrements by 1 the intensity of the fire in the agent's current location, if any, and otherwise behaves like a NOOP. Joint actions are defined as the aggregation of the agents' local actions. Movement actions are independent (agents can occupy the same location), but FF actions are not: the intensity of a fire only decreases by 1 even if multiple agents simultaneously fight it. The joint reward for the agents in a state prior to reaching T is the negative sum of the fires' intensities in that state. When the time horizon is reached, the problem episode ends, and the joint reward is 10 times the negative sum of the remaining fires' intensities, encouraging the agents to put all the fires out before the deadline.

An example of how agents might have local models of this joint model is the following. An agent's local state consists of its location, the locations and intensities of the fires, and the current delay in each cell. That is, it does *not* include the positions of other agents. Hence, its local action space only includes its 6 actions, and its local transition model will only model how its local actions affect

[1] What follows can be extended to local partial observability with the usual impacts on complexity.

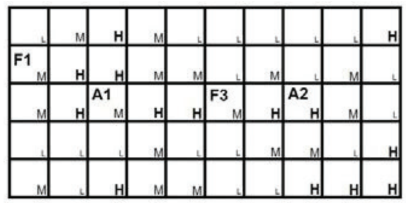


Fig. 1: Example initial state of a 10×5 firefighting grid world. Ai is the position of agent i, and Fx indicates that there is a fire in that cell with intensity x. H, M, and L designate a high, medium, or low delay respectively in that cell.

its local state. Its local reward function is the same as the global reward function; note that in this case the sum of the agents' local rewards will overestimate the true (negative) reward. Its local finite time horizon is identical to the global finite time horizon, and its local initial state distribution is calculated by directly mapping the initial distribution of global states into the local state space. Given such a local model, each agent will formulate a local policy that would fight the fires optimally if the agent were alone in the world. Note that, in general, the joint policy formed by the combination of these optimal local policies will not itself be optimal. For example, in Figure 1, both agents will be drawn to the high intensity fire first and try to redundantly fight it rather than dividing up to fight the two fires concurrently.

3 Organizational Influence in Decision-Theoretic Agents

In [4], we demonstrated how the components of the Dec-POMDP model provide a way to systematically enumerate the dimensions of the organizational design space, at least for designs intended for decision-theoretic agents. Formally, let an organizational design be defined as $\Theta = \langle \theta_1, \cdots, \theta_n \rangle$ where $\theta_i = \langle S_{\theta_i}, \alpha_{\theta_i}, A_{\theta_i}, R_{\theta_i}, P_{\theta_i}, T_{\theta_i} \rangle$ is the local organizational model for agent i. θ_i specifies the local state space, initial state distribution, action space, reward function, transition function, and finite time horizon (FTH) for agent i, when the agent is performing its role in organization Θ. We make no restricting assumptions about symmetry among the agents' organizational specifications, but rather allow for arbitrary social structure that establish various roles and responsibilities (e.g., such as in [6, 7] among others), which are then expressed via the Dec-POMDP components. We now step through each of the components and discuss how each could be used to inject organizational influences.

Rewards: The idea of modifying local models to improve coordination is not new. In particular, a growing body of literature on *reward shaping* specifically looks at how agents' reward functions can be manipulated to bias agents into taking actions that benefit the collective [8, 9]. For example, reward shaping can lead an agent to establish conditions that have no (unshaped) local reward, but that enable other agents to then take actions that lead to high joint reward.

Hence, one obvious dimension in the organizational design space is the space of alternative combinations of reward functions to assign to agents.

Transitions: While an organizational reward function can bias an agent's actions, it cannot inform other agents that those actions are now likely to be performed. For example, consider the situation where one agent can establish a condition that enables another to take actions which ultimately lead to high reward. The second agent might not perform necessary precursor actions because its local model indicates that the condition is unlikely to be established by default. The organization could give the second agent a modified transition function indicating that, given organizational influences elsewhere, the condition of interest is now more likely to be established. Hence, besides reward functions, transition function modification is another dimension of organizational design.

Actions: Without specialized optimizations during policy creation, organizational shaping of reward and/or transition components will not reduce the size of the agents' local policy spaces, but only their decisions about which of those policies are optimal. Redesigning some of the other components of an agent's decision model, however, can achieve another objective often attributed to organizational influence, which is to simplify an agent's reasoning. For example, the organizational design might associate different roles with different agents and thus induce agents to specialize in the possible actions they perform. The design can give an agent a reduced action specification that constrains its choices in some (or all) states. Chosen well, such restrictions not only help agents pursue complementary policies, but simplify planning for each. Like reward shaping, encoding organizational influence as constraints on behavior is a familiar approach in the literature [2].

States: In a factored state representation, the organizational design could recognize that there are features that an agent can sense that are unnecessary to represent given the organization. In our running firefighting example, for instance, the organizational design might capture that some (distant) fires need not be modeled by an agent at all (because they are the responsibility of other agents), thereby simplifying its local decision problem. Further, the organizational design might purposely augment an agent's local state representation with new features, where the design recognizes that those features are crucial to distinguishing between states that otherwise would look locally identical. For instance, in our running example, to improve coordination the design might require that each agent tell the others which fire it is now working towards extinguishing. Such augmentations must be done with caution, however, as we will see in Section 4.3.

Initial State and FTH: Finally, an organization can also influence an agent's behavior through α_{θ_i} and T_{θ_i}. In the firefighting scenario, an organization could, for example, initially position the firefighters at particular locations and reflect the influence on initial state correspondingly. Similarly, by shaping the rewards, transitions, and actions of the various agents, the organizational design might determine that the improved parallelism from coordination means that agents can safely reason over shorter time horizons. Alternatively, the design might improve coordination by increasing T_{θ_i} for the agents, effectively asking them to be less myopic.

3.1 Empirical Results

We now briefly summarize aspects of our experimental evaluation given in [4], which uses the problem formulation already described (Section 2), in terms of state features, agents' actions, their transitions, and joint reward function, but uses 10 cooperative agents and 10 fires on a 25×10 grid. Fires are distributed uniformly randomly over the entire grid, with intensities drawn uniformly from $\{1, 2, 3\}$. The experiments from [4] did not model delay (i.e. $\forall c, \delta_c = 0$), however it is used in our new experiments (Sections 4, 5). As we discovered in our initial experiments, the initial locations of the agents can favor, or disfavor, some organizational designs. Thus, in our experiments, we consider two extreme cases of initial locations for the agents: where they are evenly spread around the environment; and where they are clustered at the center of the grid. These environments represent the best and worst case for the baseline local behavior respectively.

To understand the impact of designing along different dimensions, we implemented largely the same organizational structure using the different model components. Each organization assigns each agent its own primary and secondary region of responsibility – in which agents are responsible for fighting fires – by exerting influence in the designated model component (e.g., **ActionOrg** influences the agents' local action spaces, **RewardOrg** their local rewards, etc.). The **Local Baseline** does not exert any organizational influence, while **FullOrg** exerts influence over every model component.

To test the degree to which an organizational design provides long-term benefit to a multiagent system, we ran each of these fixed organizational designs over a large number of randomly-generated problem instances, where each instance is an episode that begins with a randomized configuration of fires and ends when the time horizon is reached. By the luck of the draw, some problem instances might be well suited to one organization over another. We focus on aggregate performance over many episodes not only to smooth out the randomness of the instances but moreover to identify an organization's effectiveness over the long term, due to the assumption that organizational design has a high cost that is amortized over time. A well-designed organization is one that improves long-term joint reward while also simplifying each agent's local planning problem.

Table 1 summarizes results of some experiments and confirms intuitions. As others have discovered, reward shaping can be a powerful tool for increasing the expected joint reward; however, it does not generally reduce the agents' computational efforts. Shaping the transition functions can also yield a large increase in the expected reward, but can substantially increase the agents' computational costs. We also observe that constraining the agents' action or state spaces can greatly simplify the agents' decision problems and also increase the expected joint reward. Finally, with **FullOrg**, we observe that the organizational influences in the components are not completely redundant, as it is largely possible to

Table 1: Mean results for expected reward and CPU time to create policy (ms).

	Initially Spread		Initially Clustered	
	Reward	**CPU Time**	**Reward**	**CPU Time**
Local Baseline	-107.40	1646	-436.7	10912
RewardOrg	-91.45	1817	-242.0	11051
TransitionOrg	-85.14	14606	-222.5	10859
ActionOrg	-94.14	551	-264.5	621
StateOrg	-94.14	1237	-254.4	1588
FullOrg	-87.51	5476	-250.4	2652

obtain the additive benefits of each of the other organizations. While the above highlights our results, more details and explanations can be found in [4].

4 Factored Framework

The organization instantiations just described were formed by replacing one or more components of the agents' local models with components provided by the organizational design. While doing so has demonstrated effectiveness, as previously argued it risks "throwing out the baby with the bath water" by potentially overwriting hard-gained expertise of an agent. Our solution to this problem is to factor both the environmental model and the organizational specification. By doing so, the organizational specification can include (and overwrite) only those factors related to inter-agent coordination, and leave alone factors associated with agents' local expertise. By overlaying the organizationally-specified factors on top of its local model, an agent can use the augmented model to make its more subtly organizationally-influenced local decisions.

4.1 Factoring the Local Models

Factoring local MDPs is not a new idea (e.g., [10, 11] among others), and provides a method to exploit the independencies within the model's structure to reduce the effective size of the decision problem and its representation. Here, however, our motivation for factoring the local MDP is to divide the model into pieces so that an organizational design can influence only those pieces related to inter-agent coordination. This leaves the remainder of the model untouched, and allows an agent to utilize its local expertise for completing tasks.

We adopt a common factoring scheme and define the local model for agent i as $\mathcal{M}_i = \langle S_i, \alpha_i, A_i, R_i, P_i, T_i \rangle$, where:

- S_i is the finite set of local states. Local state is factored into m_S factors, $S_i = F_{i_1} \times F_{i_2} \times \cdots \times F_{i_{m_s}}$ where F_{i_j} is the domain for local state factor j.
- $\alpha_i = \langle \alpha_{i_1}, \cdots, \alpha_{i_{m_\alpha}} \rangle$ where $\alpha_{i_j} : (\times_k F_{i_k}) \to [0,1]$ specifies the jth component of m_α local initial state distribution factors. Assume each state factor appears in exactly 1 α_{i_j} (i.e. the state factors are partitioned across the α_{i_j}'s).

– A_i is the finite set of m_A local actions, $A_i = \{a_1, \cdots, a_{m_a}\}$. Assume that all actions are primitives, and thus they cannot be factored further.
– $R_i = \sum_{j=1}^{m_R} R_{i_j}$ where $R_{i_j} : (\times_k F_{i_k}) \times A_i \rightarrow \mathbb{R}$ is the jth factor of m_R local reward function factors. Unlike with the initial state distribution, a state factor may contribute to multiple local reward factors.
– $P_i = \langle P_{i_1}, \cdots, P_{i_{m_P}} \rangle$ where $P_{i_j} : (\times_k F_{i_k}) \times A_i \times (\times_{k'} F_{i_{k'}}) \rightarrow [0, 1]$ specifies the jth factor of m_P local transition function factors. Assume that each state factor appears in the target of exactly 1 P_{i_j} (the k' indices), but may appear in the source of multiple P_{i_j} (the k indices).
– $T_i \in \mathbb{R}$ is the finite time horizon.

Figure 2 presents an example $\mathcal{M}_i$ for the firefighting grid world domain at a time t as a dynamic Bayesian network (DBN), a common tool for depicting factored MDPs. The local state factors are time ($TIME$), the agent's current position (POS), the intensity of fires in each of the C cells (INT_c for cell c), and the delay conditions in each cell (δ_c for cell c). The fire intensity factors together with the time determine the local reward factors, each of which is the negative intensity of the fire at that location and 10 times that amount if the time is equal to the finite time horizon. There are four local transition factors, the first (P_{i_0}) increments the time every step. The second (P_{i_1}) decrements a fire's intensity if the agent performs the fight fire action in that cell. The third (P_{i_2}) changes the agent's position depending on the agent's current position, action, and delay conditions. Finally, the fourth (P_{i_3}) updates delay conditions in each cell.

4.2 Factoring the Organization

We now turn to factoring the organization. As with the local models, we want to factor the organization into pieces so that the organizational design only impacts those factors related to an agent's role in inter-agent coordination. Unlike with the local models, however, here we need to express the changes being made to the model as opposed to an entire model. There are three types of changes an organizational design might wish to convey: changing a factor, for example altering a transition factor to account for other agents being responsible for fighting fires in some geographic regions; adding a new factor, for example a new reward factor for being located in a region of responsibility; or blocking a factor, for example ignoring distant fires' intensities in the state features. Figure 3 shows this particular organization for the running firefighting example. Shaded regions indicate factors that are organizationally added or altered, while lightly dotted regions indicate factors that are organizationally blocked (and thus ignored).

Our organizational specification includes two sets of factors: those to be added (or altered) and those to be blocked. We do not need to explicitly distinguish factors that are replacing existing factors from those that are completely new, since the organization is overlaid on top of the agent's existing model (discussed in Section 4.3). We formally define an organization $\Theta = \langle \theta_1, \cdots, \theta_n \rangle$, where θ_i is the organizational component for agent i, defined as $\theta_i = \langle \{F_{i_j}^{\Theta}\}, \{\bar{F}_{i_j}^{\Theta}\}, \{\alpha_{i_j}^{\Theta}\}, A_i^{\Theta}, \bar{A}_i^{\Theta}, \{R_{i_j}^{\Theta}\}, \{\bar{R}_{i_j}^{\Theta}\}, \{P_{i_j}^{\Theta}\}, T_i^{\Theta} \rangle$, where:

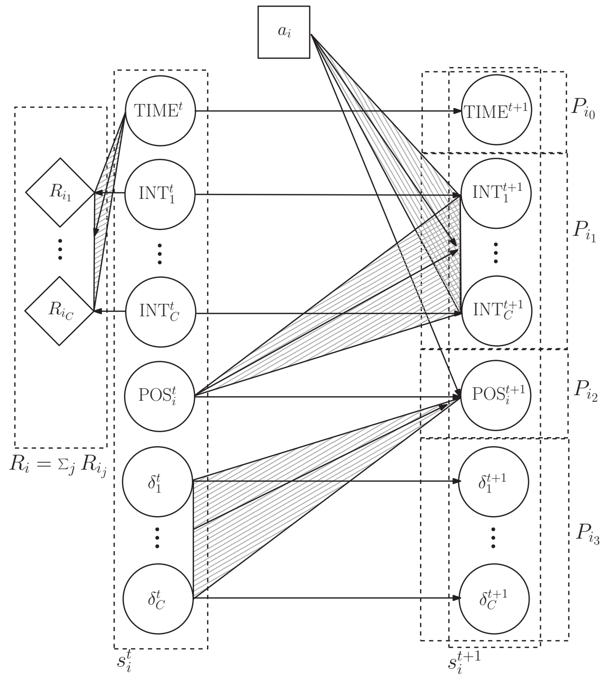


Fig. 2: An example model for agent i for the firefighting grid world domain at a time t represented as a dynamic Bayesian network.

- $\{F^{\Theta}_{i_j}\}, \{\bar{F}^{\Theta}_{i_j}\}$ specify the sets of local state factors organizationally added and blocked from the agent's model respectively. We assume that these two sets are consistent such that no factor appears in both sets, $\{F^{\Theta}_{i_j}\} \cap \{\bar{F}^{\Theta}_{i_j}\} = \emptyset$.
- $\{\alpha^{\Theta}_{i_j}\}$ is the organizational augmentation for the local initial-state distribution.
- $A^{\Theta}_i = \{a^{\Theta}_{i_j}\}, \bar{A}^{\Theta}_i = \{\bar{a}^{\Theta}_{i_j}\}$ specify the sets of local actions organizationally added and blocked respectively. We assume that these two sets are consistent such that no action appears in both sets, $A^{\Theta}_i \cap \bar{A}^{\Theta}_i = \emptyset$.
- $\{R^{\Theta}_{i_j}\}, \{\bar{R}^{\Theta}_{i_j}\}$ specify the sets of local reward factors organizationally added and blocked from the agent's model respectively. We assume that these two sets are consistent such that no factor appears in both sets, $\{R^{\Theta}_{i_j}\} \cap \{\bar{R}^{\Theta}_{i_j}\} = \emptyset$.
- $\{P^{\Theta}_{i_j}\}$ is the organizational augmentation for the local transition function.
- T^{Θ}_i is the organizational finite time horizon.

Note that $\{\alpha^{\Theta}_{i_j}\}, \{P^{\Theta}_{i_j}\},$ and T^{Θ}_i do not have "blocked" counterparts like the other components. This is because an agent must always have a model of this information or its decision-making process is under-defined. For example, each state factor must be included in exactly one transition factor, otherwise, what should happen to it upon taking an action? It could be treated as a constant and not change; however, this effectively defines a transition function–as does

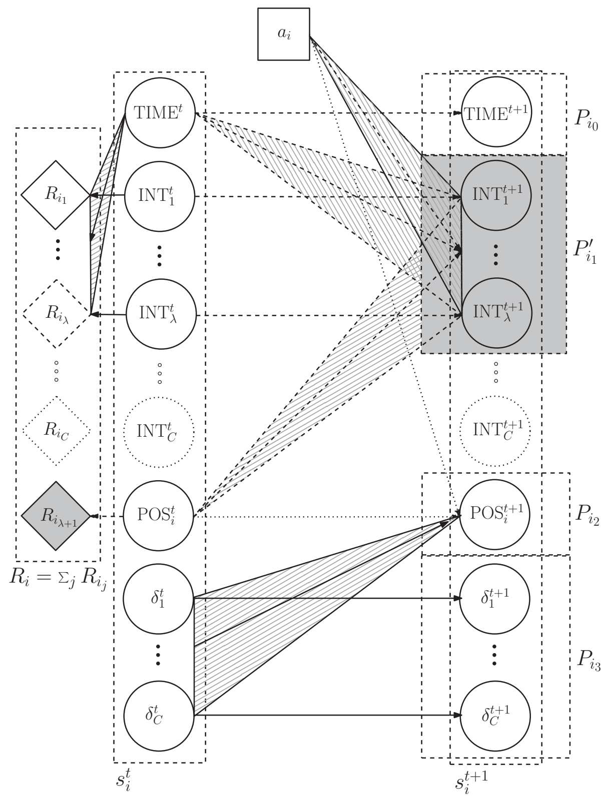


Fig. 3: An example organizational augmentation to the DBN from Figure 2. Shaded regions indicate factors that were organizationally altered or added, while dotted regions indicate factors that were organizationally blocked.

any other "default" solution. Therefore, if an organizational design wishes to block a local transition factor, it should also include a replacement transition factor to prevent the agent's augmented model from becoming under-defined. This is equivalent to altering the transition factor, and thus being able to block a transition factor is unnecessary. $\{\alpha_{i_j}^{\Theta}\}$ and T_i^{Θ} have analogous reasoning for not including their "blocked" counterparts.

4.3 Overlaying the Organization onto the Agents' Models

In general, each agent overlays its organizational model, θ_i, onto its pre-existing local model, $\mathcal{M}_i$, by following organizational influences where provided, and otherwise utilizing its local factors. This methodology stems from the philosophy that the organizational design represents a more globally-aware viewpoint than the individual agent has (although need not be a complete global view[2]). As such, the organizational design process should better know which factors should be organizationally influenced, and the agents should abide by its influences. However, we also point to work on organizationally adept agents [4, 12], in which the agents reason about the effectiveness and appropriateness of their organization, and revise their organization to be better suited to their current environment, where this overlay process is overruled by dynamic adjustments.

Procedurally, each agent independently overlays its received organizational specification onto its local model to create its augmented model as follows. If the organization specifies a factor, then the agent uses it (replacing any corresponding local factor as necessary). If the organization blocks a factor, then the agent ignores the corresponding local factor and does not use it during its planning process. Finally, if the organization does not specify anything about an agent's local factor, then the agent uses its local factor.

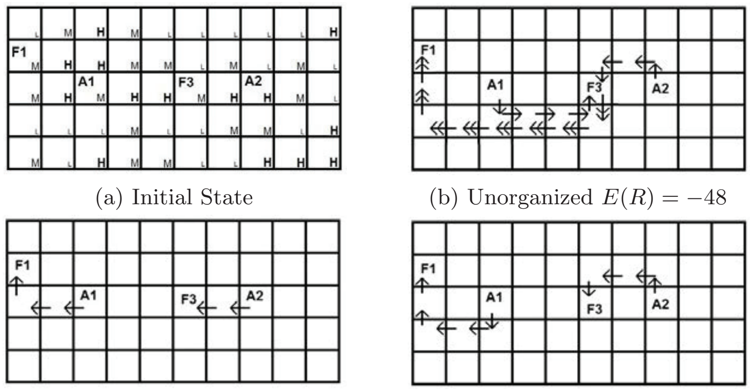
Since we cannot assume that an entity creating the organizational design has complete knowledge about any of the agents (otherwise it could redundantly specify the agents' local expertise in Θ), it is possible that the organizational design blocks a factor that the agent does not have a model of. Such circumstances are easily resolved, however, since the organization is essentially blocking a factor from being modeled that wasn't being locally modeled anyway.

However, it is also possible for the organizational design to add a new factor that an agent is incapable of modeling (e.g., due to limitations in its sensor or actuator capabilities). For example, in the firefighting domain, an organizational design could indicate that an agent must model the current wind direction by including it within $\{F^{\Theta}_{i_j}\}$. However, if the agent does not have a sensor to measure the wind direction, then how should it proceed? One naïve option would be to simply disregard that factor of the organizational specification. However, this is prone to a cascading effect, for example when reward and transition factors are dependent on state factor values, and could create inconsistencies among the organizational influences being exerted. Another option, which we assume here instead, is that the agent must inform the process providing the organizational design that it is incapable of modeling certain factors. The organization can then be redesigned to not include those factors. This procedure can iterate until a model-able design is devised, thus ensuring that the organization as implemented by the agents remains in a consistent and complete state. It remains an open question as to how this procedure might be efficiently performed or if it is assured that a model-able design can be converged upon.

5 Evaluation

We return to our firefighting example to illustrate the advantages of an organizational design selectively overwriting parts of components using the factored

[2] We are agnostic about the specifics of whom creates these influences (e.g. it could be a supervisor, peer, distributed process, etc.) provided the design stems from a more global viewpoint than the agent possesses.



(a) Initial State

(b) Unorganized $E(R) = -48$

(c) Unfactored Org. $E(R) = -33$

(d) Factored Org. $E(R) = -28$

Fig. 4: Policies created by each of our frameworks in the given initial state and the expected reward, E(R), obtained from executing those policies.

framework. Figure 4a repeats the simple firefighting problem given in Figure 1 including the current delay status of each location which is known by both agents. Figure 4b portrays the behaviors of the agents without any organizational design: they both (redundantly) fight the high intensity fire, and then both (redundantly) fight the other fire (expected reward $E(R) = -48$). Figure 4c shows what happens with an unfactored organizational design, where the design uses long-term average delays in the transition function it gives to agents, overwriting the agents' better knowledge of the current delays in each location ($E(R) = -33$). Finally, Figure 4d shows the paths agents take with the factored organizational design, replacing parts of agents' transition models to help agents avoid redundancy, but preserving parts of agents' local transition models that encode current delay statuses ($E(R) = -28$). The resulting behaviors look indirect, but follow low-delay paths that in expectation are faster, allowing agents to achieve 15% higher $E(R)$.

Our primary contribution of this paper is to provide a methodology for an organization to augment, rather than replace, the agents' pre-existing knowledge, skills, and expertise. As just illustrated, our factored model achieves this goal by allowing an organizational design to only specify the model factors related to inter-agent coordination, and to defer to the expertise of agents themselves regarding details of task execution. However, a more thorough evaluation requires examining the effectiveness of our factored framework in several other distinct dimensions. In particular, some dimensions of interest include: expressive power; expected reward; specification compactness; organizational design space size; and ease and flexibility of use for agents. We are embarking on trying to perform quantitative evaluations on some of these dimensions. As a preliminary step towards this, in what follows we provide several qualitative insights drawn from conceptual analysis and intuitive reasoning.

Expressive Power: Since the organizational design could include every factor in the agents' models within the organizational specification but could also specify only a subset of those factors, the expressive power of our factored framework is strictly greater than that of the unfactored framework. That is, all organizations that could be specified with the unfactored framework can be specified in the factored framework; however, the converse is false.

Expected Reward: The difference in expected reward from using our factored framework as compared to our unfactored framework is dependent on the problem domain. Specifically, the expected reward from using our factored framework will increase by the exact amount that having the additional local agent knowledge would increase the expected reward. This amount can vary drastically from not mattering at all (i.e. the organizational design process knew everything about the task environment and inter-agent behaviors) to being crucially essential (i.e. the organizational design process knew nothing). In general, however, we expect that most domains will lie between these extremes; for example, the problem in Figure 4 attains a modest 15% improvement thanks to the factoring.

Specification Compactness: Since an organizational design need only specify the factors that it intends to influence (which is a weak subset of the entire organizationally augmented model), it is clear that the organizational specification size will be no larger than that of the unfactored framework. The exact amount of reduction will be largely determined by the proportion of factors being organizationally influenced; however, it is worth noting that there will likely *not* be a linear correspondence between proportion of factors being influenced and proportion of total specification size relative to the unfactored framework. That is, the factored framework inherits the typical benefits of a factored model whereby independence between factors can greatly reduce the specification size. As such, even if the factored organizational model includes a large proportion of the factors, it is possible that the total specification size is still greatly reduced as compared to the unfactored framework. The degree of specification size reduction is thus not only dependent on the proportion of factors being influenced, but also upon the degree of coupling between those factors and the remainder of the model. Environments with tightly-coupled factors will result in larger specification sizes (although no larger than the unfactored specification), while environments with loosely-coupled factors will yield smaller specification sizes.

Organizational Design Space Size: Since we make no assumptions about how many or which factors will be organizationally influenced, the organizational design space can become exponentially larger in the worst case. That is, in addition to specifying appropriate values for the factors being influenced, the organizational design process must now also decide which factors should be included in the organizational specification. In the worst case, this means the organizational design space includes an organization for every possible subset of factors. However, if the subset of appropriate factors is (heuristically) known, then the organizational design space could be greatly reduced due to only having to design those factors being included in the organizational specification. In general, every factor that is not included in the specification will yield an exponential reduction in the organizational design space size; however, as with the specification size reduction, the exact degree of organizational design space reduction also depends on the degree of coupling between the factors. Quantitatively evaluating the precise effects on the organizational design space thus remains an open challenge due the need to first create a more algorithmic method for designing organizations for decision-theoretic agents.

Usage by Agents: The methodology presented in Section 4.3 is clearly a more complex process for the agents than simply using a new, complete local model supplied by the organization. Each agents is now required to overlay a subset of factors onto its existing local model, and to reason about if those factors align with what it is internally capable of modeling. However, this increased complexity also allows for greater flexibility. For example, using the factored framework allows an agent to selectively adopt/reject portions of the organizational design. With organizationally adept agents (OAAs) [4, 12], the agents could decide to change or abandon only part of their organization (e.g., the portions that are not functioning as anticipated), leaving other parts untouched. Such adaptations must be done with care, however, as it is also possible that this could create a cascading effect, such as if the organizational design has correlated factors, but only one of those factors is altered by the OAAs. Quantitatively evaluating the benefit of this flexibility thus remains an open challenge since the current OAA mechanisms to agree on modifying their organization need to be extended to reason about issues of interdependencies among different factors.

6 Related Work and Conclusions

Much of the literature in multiagent organization design and specification concentrates on formulating organizational modeling languages (OMLs), such as MOISE$^+$ [6] and OMNI [7], among a variety of others. Though the specifics of these OMLs vary, they generally emphasize specifying an agent organization at an abstract level in terms of roles, role relationships/interactions, norms, etc. They also tend to be agnostic about how an agent would map the abstract specification into its internal reasoning processes. Hence, our work here complements the OML research, helping to bridge the gap between modeling and implementation by identifying opportunities and limitations in social structures that can be meaningfully mapped into influences over decision-theoretic agents.

In another line of work, there has been extensive research on factoring decision-theoretic models [10] and efficient solution algorithms for finding policies in factored models [11]. While our work utilizes that research as a baseline for factoring the agents' local models, we extend that framework by incorporating organizational influences. As was the case for OMLs, our work can be seen as complementary to the factored MDP work, bridging the gap between low-level implementation research and the higher-level organizational models.

In this paper we have briefly described our previous findings [4] in injecting organizational influences within the various model components of a decision-theoretic agent. We intuitively described and empirically demonstrated how influencing the Dec-POMDP components can both increase the agents' expected joint reward as well as simplify their local decision problems as compared to a baseline local model. We then extended that prior research to create a factored framework where the organizational designer only specifies those factors related to inter-agent coordination, and allows each agent to retain its pre-existing local expertise for solving the local aspects of its problems. Finally, we presented a qualitative evaluation of our factored framework across several performance dimensions for both the organizational design process and the agents. In the future, we plan to more precisely evaluate the quantitative performance of our factored framework; however, doing so first requires a more precise and algorithmic notion of the organizational design process and OAAs. Additionally, we plan to extend an agent's overlaying process to incorporate and balance influences from multiple organizations in which it might be simultaneously participating.

7 Acknowledgments

We thank the anonymous reviewers for their thoughtful comments, and our collaborators at the University of Massachusetts for their consistently helpful feedback. This work was supported by NSF grant IIS-0964512.

References

1. So, Y.p., Durfee, E.H.: Designing organizations for computational agents. In: Simulating Organizations. MIT Press, Cambridge, MA, USA (1998) 47–64
2. Fox, M.S., Barbuceanu, M., Gruninger, M., Lin, J.: An organizational ontology for enterprise modeling. In: Simulating organizations. MIT Press, Cambridge, MA, USA (1998) 131–152
3. Corkill, D.D., Lander, S.E.: Diversity in Agent Organizations. Object Magazine **8**(4) (1998) 41–47
4. Sleight, J., Durfee, E.H.: A decision-theoretic characterization of organizational influences. In: AAMAS. (to appear 2012)
5. Bernstein, D.S., Givan, R., Immerman, N., Zilberstein, S.: The complexity of decentralized control of Markov decision processes. Mathematics of Operations Research **27**(4) (2002) 819–840
6. van Riemsdijk, M.B., Hindriks, K.V., Jonker, C.M., Sierhuis, M.: Formalizing organizational constraints: A semantic approach. In: AAMAS. (2010) 823–830
7. Vázquez-Salceda, J., Dignum, V., Dignum, F.: Organizing multiagent systems. JAAMAS **11** (November 2005) 307–360
8. Wolpert, D., Tumer, K.: Optimal payoff functions for members of collectives. Advances in Complex Systems **4**(2/3) (2001) 265–279
9. Agogino, A.K., Tumer, K.: Multi-agent reward analysis for learning in noisy domains. In: AAMAS. (2005) 81–88
10. Boutilier, C., Dearden, R., Goldszmidt, M.: Stochastic dynamic programming with factored representations. Artificial Intelligence **121** (2000) 49–107
11. Guestrin, C., Killer, D., Parr, R., Venkataraman, S.: Efficient solution algorithms for factored mdps. JAIR **19** (2003) 399–468
12. Corkill, D., Durfee, E., Lesser, V., Zafar, H., Zhang, C.: Organizationally Adept Agents. In: COINS2011 Workshop at AAMAS. (2011)