Organizational Design Principles and Techniques for Decision-Theoretic Agents

Jason Sleight and Edmund H. Durfee
Computer Science and Engineering
University of Michigan
Ann Arbor, MI 48109
{jsleight,durfee}@umich.edu

ABSTRACT

Recent research has shown how an organization can influence a decision-theoretic agent by replacing one or more of its model components (transition/reward functions, action/state spaces, etc.), and how each of these influences impacts the agent's decision-making performance. This paper delves more precisely into exactly which parts of an agent's model should be organizationally influenced, and asserts a broader principle for delineating what aspects of an agent's behavior an organization should be sanctioned to influence. We present a formal framework for specifying factored organizational influences and incorporating them into agents' decision models, and empirically demonstrate that organizational specifications based on our proposed principle outperform the alternatives. We further describe an algorithm for automating the organizational-design process that is inspired by this principle, and demonstrate empirically that its organizational designs are both intuitively sensible and also find and exploit domain structure that our hand-generated designs miss.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Coherence & Co-ordination, Multiagent Systems*

General Terms

Design, performance

Keywords

Organization, organizational design

1. INTRODUCTION

A well-designed organization judiciously applies its available levers of influence to lightly, but firmly, guide its members into working well together. For decision-theoretic agents, the available levers of influence correspond to shaping the components of the agents' decision-theoretic models, including their reward and transition functions and their state and action spaces (e.g., [1, 11, 14, 16]). Firm but light guidance means that components should be chosen and shaped by the organization to help agents avoid doing (and preferably even avoid thinking about doing) redundant, contradictory, or counterproductive actions, while leaving them freedom to exercise their abilities at their own discretion otherwise.

Recently [11], we posited that the components of a decision-theoretic model define a vocabulary in which organizational designs can be expressed to decision-theoretic agents, and examined how organizationally influencing the agents via different (combinations of) components impacts the quality and costs of coordinated decision making. Our work demonstrated how replacing components with organizationally-provided ones could firmly guide agents' behavior in effective ways. We contend, though, that this process of replacing entire components was also unnecessarily heavy-handed.

In this paper, we factor the components of the decision-theoretic model to create a richer vocabulary in which organizational design can more flexibly be expressed. A contribution of this paper (Section 3) is a description of this vocabulary, along with the semantics attached to it by an agent incorporating a design into its local model. Unfortunately, the richer vocabulary also induces an exponential growth in the space of expressible organizations, which can bog down the design process. To combat this growth, we also contribute a strategy for forming organizational designs based on the principle that organizational influence should be limited to addressing agents' reward and/or transition dependencies. In Section 4, we provide our rationale for this principle and provide experimental evidence of its effectiveness.

Even with this design principle, however, the organizational design process can be time-consuming and error-prone if done by hand, as teasing out the pertinent interdependencies can be difficult. This argues for automated techniques for conducting the organizational design process (ODP); such techniques are the other main contribution of this paper (Section 5). Briefly, the underlying insight behind our automated technique for ODP is to also cast the ODP in decision-theoretic terms, modeling and reasoning about the agents' joint behaviors abstractly to predict desirable patterns of joint action, and then influencing the agents into these coordination patterns. Section 5 provides details of the ODP, and presents preliminary empirical results showing that it finds organizational designs that make sense intuitively and that can exploit structure in the domain. We end the paper (Sections 6 and 7) by summarizing our work and discussing how it relates to other past and ongoing work in organizational design for multiagent systems. We next (Section 2) describe our agents' decision-theoretic framework and introduce the stylized firefighting domain that we use for illustration and experimentation throughout this paper.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013), Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.*

2. PROBLEM DOMAIN

Like in our previous work [11], we adopt a standard decentralized partially observable Markov decision process (Dec-POMDP) paradigm [2], $\mathcal{M} = \langle \mathcal{N}, S, \alpha, A, R, P, \Omega, O, T \rangle$, 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 [2], 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, and utilize factored models (e.g., [4]) to more compactly represent the decision problem. For example, global state is factored into a set of m_S state features, such that $S = F_1 \times \cdots \times F_{m_S}$, where F_j is the (finite) domain of state feature j. Each agent i has a local state representation S_i consisting of a subset of the m_S features. Each X of the other model components (i.e., $X \in \{\alpha, A, R, P, \Omega, O\}$), is then defined in terms of this factored state representation, and similarly factored into m_X features. The number of factors in a model-component can differ from agent to agent, but to avoid notational clutter this subtlety is ignored in the description that follows. We further adopt the common assumption of local full observability (each agent's local observations uniquely determine its local state).

Given these assumptions, the local decision model $\mathcal{M}_i$ of agent i represents a local Markov decision process (MDP) $\mathcal{M}_i = \langle S_i, \alpha_i, A_i, R_i, P_i, T_i \rangle$, which defines the local states, actions, rewards, etc.

- $S_i = F_{i_1} \times F_{i_2} \times \cdots \times F_{i_{m_S}}$.
- $\alpha_i = \langle \alpha_{i_1}, \cdots, \alpha_{i_{m_\alpha}} \rangle$ where $\alpha_{i_j} : (\otimes_k F_{i_k}) \to [0,1]$ and α_i partitions $\{F_{i_k}\}$.
- $A_i = \{a_{i_1}, \cdots, a_{i_{m_a}}\}$.
- $R_i = \sum_{j=1}^{m_R} R_{i_j}$ where $R_{i_j} : (\otimes_k F_{i_k}) \times A_i \to \mathbb{R}$.
- $P_i = \langle P_{i_1}, \cdots, P_{i_{m_P}} \rangle$ where $P_{i_j} : (\otimes_k F_{i_k}) \times A_i \times (\otimes_{k'} F_{i_{k'}}) \to [0,1]$ and the target of P_i partitions $\{F_{i_{k'}}\}$.
- $T_i \in \mathbb{Z}^+$.

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 reuse a simplified firefighting scenario [11], 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, $\ell_i \in Cells$ for agent i; the fire intensity, $I_c \in \mathbb{Z}^+$ for each cell c. Further, compared to our prior work [11], we also add a delay, $\delta_c \in [0,1]$ for each cell c, which stochastically prevents movement into that cell with probability δ_c. Figure 1 shows an initial global state, where the locations of agents A1 and A2 are shown, along with the intensity of fire in the 2 cells with $I_c > 0$. In Figure 1, (H)igh, (M)edium, and (L)ow delay correspond to δ equal $0.8, 0.5$, and 0.0 respectively. 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 thus equates to a NOOP with probability δ_{c_dest} (or if there is no cell in that direction); and a fight-fire (FF) action that decrements by 1 the intensity of the agent's current cell (to a minimum of 0) 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 cell only decreases by 1 even if multiple agents simultaneously fight it. The joint reward for the agents in a state prior to reaching time horizon T is $-\sum_c I_c$. When T is reached, the problem episode ends, and the joint reward is $-10 \sum_c I_c$, encouraging the agents to put all the fires out before the deadline.

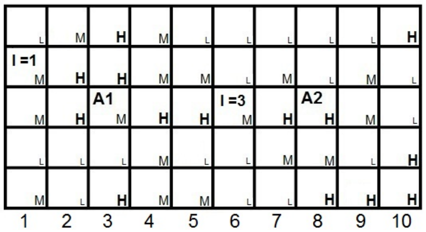


Figure 1: Example initial state in the firefighting grid world. Ai is the position of agent i, and $I = x$ indicates that there is a fire in that cell with intensity x. Letters designate a (H)igh, (M)edium, or (L)ow delay in that cell.

An example of an agent's local model from this joint model follows. An agent i's local state consists of ℓ_i, I_c for each cell, and δ_c for 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 only models how its local actions affect 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. Figure 2 shows the local model for agent i represented as a dynamic Bayesian network (DBN). 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 cell first and try to redundantly fight its fire rather than dividing up to fight the two cells with fires concurrently.

3. FACTORED ORGANIZATION

Our prior work showed that organizational influence can be exerted on decision-theoretic agents by replacing components of their decision models [11]. Our objective in this paper is to allow more nuanced organizational influence through selected *factors* of components, and so organizational designs are factored like local models. To selectively modify a local model, an organizational design should be able to express: alterations to an agent's existing factors, such as altering a transition factor to reflect expectations about fires in nearby cells being

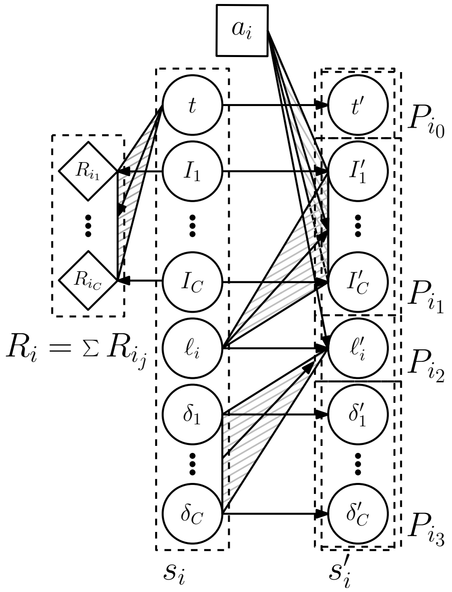


Figure 2: An example factoring for agent i represented as a dynamic Bayesian network (DBN).

extinguished (due to other agents' efforts); additions to an agent's factors, such as introducing a new reward factor to induce an agent to fight fires in a region of responsibility and punish it for fighting fires in others' regions; and deletions of an agent's factors, such as blocking an agent from including distant cells' intensities in its local state.

Our organizational specification classifies factors into two sets: those to be added and those to be blocked, where if an added factor already has a corresponding existing factor for the agent, it simply alters (replaces) it. 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 $\theta_i = \langle \{F^{\Theta}_{i_j}\}, \{\bar{F}^{\Theta}_{i_j}\}, \{\alpha^{\Theta}_{i_j}\}, \{a^{\Theta}_{i_j}\}, \{\bar{a}^{\Theta}_{i_j}\}, \{R^{\Theta}_{i_j}\}, \{\bar{R}^{\Theta}_{i_j}\}, \{P^{\Theta}_{i_j}\}, T^{\Theta}_i \rangle$, where:

- $\{F^{\Theta}_{i_j}\}, \{\bar{F}^{\Theta}_{i_j}\}$ specify the sets of local state factors organizationally added to and blocked from agent i's model respectively.
- $\{\alpha^{\Theta}_{i_j}\}$ is the organizational augmentation for agent i's local initial-state distribution.
- $\{a^{\Theta}_{i_j}\}, \{\bar{a}^{\Theta}_{i_j}\}$ specify the sets of agent i's local actions organizationally added and blocked respectively.
- $\{R^{\Theta}_{i_j}\}, \{\bar{R}^{\Theta}_{i_j}\}$ specify the sets of agent i's local reward factors organizationally added and blocked respectively.
- $\{P^{\Theta}_{i_j}\}$ is the organizational augmentation to agent i's local transition function.
- T^{Θ}_i is agent i's organizational finite time horizon.

We restrict specifications to those where the state factor, action, and reward components are consistent such that no factor appears in both sets; for example $\{R^{\Theta}_{i_j}\} \cap \{\bar{R}^{\Theta}_{i_j}\} = \emptyset$. Note that some components (α, P, T) do not have "blocked" counterparts like the other components, because an agent must always have a model of this information or its decision-making process is under-defined. Hence, an organizational design cannot block a factor in these components without replacing it, which is equivalent to altering it. We further restrict specifications to be consistent with what an agent is internally capable of modeling.

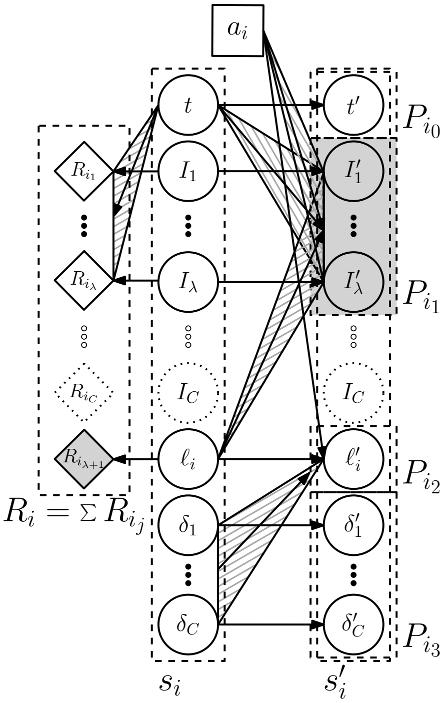


Figure 3: An example organizational augmentation to the DBN from Figure 2. Shaded regions indicate factors that were organizationally altered or added, while dotted regions indicate factors that were organizationally blocked.

When incorporating an organizational specification into its local model, an agent i overlays θ_i onto $\mathcal{M}_i$ to create its augmented model by adding entirely new local factors, removing blocked local factors, and overwriting replaced local factors. This overlaying process thus resembles how, for example, coordination locales model domain dynamics by overriding an agent's local transition/reward models, and social model shaping augments those local models to coerce coordination [1, 14, 16]. Figure 3 shows an example organizational transformation to the DBN from Figure 2, where agent i is assigned responsibility for cells 1 through λ. The organization specifies this responsibility by blocking I_c's outside of the agent's region, adding a new reward factor for being located within its region, and modifying the I_c-transition factor to account for I_c's in its region decreasing over time (due to other agents' efforts).

In this paper, we will assume that an organization, Θ, is fixed, and thus the agents will reuse it over a series of problem episodes sampled from a distribution, even though the influences might be suboptimal for some episodes. Depending on the context, however, an agent i might be permitted to disregard some (or all) of θ_i, and rely instead on its local model, or could even try θ_i, keep the useful portions, and disregard (or replace) the rest. Organizationally-Adept Agents [3], for example, could make such decisions. Alternatively, the organizational design process might police itself to only specify factors that support coordination more generally without micromanaging. In the remainder of this paper, we assume agents will adopt Θ as given, and so focus on the question of which factors Θ should specify, and the values those factors should take, to provide effective influences.

4. DESIGN PRINCIPLES

The organizational syntax we defined in Section 3 is capable of specifying much more than organizational influences, and in actuality is a general-purpose programming language for decision-theoretic agents. For example, a syntactically correct Θ could completely overwrite an agent's entire local model down to the smallest detail of how tasks are performed, the epitome of micromanagement. However, such usage would exceed the commonsense bounds of what organizations are customarily expected to influence. A natural question, therefore, is whether we can define explicit principles that embody an intuitive understanding of how organizations should be designed, and that apply to our new, richer vocabulary for factored influence on decision-theoretic agents. In this paper, we answer this question by proposing and testing one such principle to guide decisions about which portions of the agents' models an organizational design should influence.

Organizations should not dictate or micromanage because individual agents might (and generally do) possess their own expertise, and leaving them room to exercise their local expertise benefits the collective organizational objectives. Assuming that agents are locally skilled, then, we observe that what an organizational view has that individuals lack is a more global awareness across agents' activities, where if individual agents had such awareness they could make more informed decisions. Hence, an organizational design should use its more global perspective to influence agents into acting like they would if they were more globally aware themselves.

Restated, the claim is that an organizational design should use its global perspective to improve agents' decisions that impact each other, and otherwise should allow agents to exercise their local capabilities. We codify this assertion in the following **organizational design principle**: a well-designed organization should influence only the factors of agents' models that are associated with agent interactions. This principle is surprisingly applicable for creating organizational designs for decision-theoretic agents, where factors associated with interactions are directly captured in joint reward/transition functions, and indeed where the specification of agents' decision models often explicitly separates out the dependent factors (e.g., coordination locales [14, 16]) from the independent ones. This is neatly illustrated in our specification of the firefighting domain, where agent movements are independent (where one agent moves does not affect the states/rewards of another agent), but firefighting actions are not (one agent's states/rewards are affected by another's fighting of a fire).

4.1 Evaluation Strategy

To test this organizational design principle, we enumerated a space of factored organizations, each of which draws on the same, fully-specified organization, but adopts a different subset of factors taken from that full specification. We use our previously developed smallOverlapOrg [11] as our fully-specified organization in these experiments, which, roughly speaking, assigns an overlapping primary area of responsibility (PAR) to each agent. For example, in Figure 1, agent 1's PAR is the cells in columns 1–7, and agent 2's PAR is the cells in columns 4–10. The smallOverlapOrg's state component fully overwrites an agent's state space to block factors for I_c's outside the agent's PAR (i.e., agents ignore fires outside of their PARs). Its action component overwrites an agent's action space to block the agent from wasting time considering actions that would cause it to leave its PAR. The smallOverlapOrg's transition component replaces an agent's model of what its movement, firefighting, and NOOP actions do with what the organizational design thinks that they do (including capturing probabilistic changes to I_c's in the agents' overlapping PARs caused by the other agent).

Previously [11], we explored the effects on group performance of including various combinations of these components. Here, we further subdivide the components into their factors. In particular, in its factored form, the transition component has one factor for the effects of agent movement actions, and another for firefighting actions. We draw on our organizational design principle to combine only the factors that correspond to ***reward/transition-dependencies (R/T-Ds)***. Revisiting the smallOverlapOrg's components, R/T-D factors include: the entire state and action components (since these reflect the organization's global awareness that agents can depend on each other to fight fires in their respective PARs); and the transition factor for I_c's (summarizing expectations of when fires will be extinguished by other agents in the overlapping PARs). On the other hand, the transition factor for agents' movement actions is a non-R/T-D factor, because agent movements are independent.

Our organizational design principle thus leads us to the hypothesis that a specification including only these R/T-D factors will capitalize on the global perspective of an organization without overstepping the bounds into micromanagement: it will perform as well or better than other combinations of factors. To test this hypothesis, we constructed this **R/T-DOrg** organizational specification, as well as an organization that includes only non-R/T-D factors, the **non-R/T-DOrg**. For completeness, we also considered the full set of factors yielding the **unfactoredOrg** (identical to the unfactored smallOverlapOrg), and the empty set of factors which yields the **localOrg** (where agents are uninfluenced by any organization). To exercise the assumption that agents in an organization can contribute expertise of their own, we evaluated each of these four organizations across a spectrum of settings where we varied the relative quality of agent expertise compared to the organization's model. Specifically, we degraded the organization's view of the cell delays by applying a smoothing filter over the true environment delays according to the particular settings of that experiment. For example, the 100-smoothed setting had a smoothing filter iteratively applied to every cell 100 times, whereas in the 0-smoothed setting the organization's view precisely matches the real delays, etc. In this way, as the organization's model of delays blurs, it remains accurate in terms of cells' mean delays but loses precision (about the physical distribution of delays).

4.2 Results

To test the degree to which an organizational design provides long-term benefit to a multiagent system, we run each fixed organizational design over 3000 randomly-generated problem instances, where each instance is an episode that begins with a randomized configuration of cell intensities and delays, 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 the episodes not only to smooth out the randomness of the instances but also to assess an organization's effectiveness over the long term. The performance measures of interest are the expected joint reward and the

% Results Included	100%			Top 25%			Top 5%		
# Smoothing	0	10	100	0	10	100	0	10	100
R/T-DOrg	1.45%	1.40%	1.40%	6.05%	5.92%	5.98%	23.27%	23.55%	23.44%
unfactoredOrg	1.45%	-6.24%	-7.30%	6.10%	-9.61%	-10.70%	23.21%	-19.48%	-19.76%
non-R/T-DOrg	0.01%	-7.41%	-8.29%	0.00%	-12.26%	-13.02%	0.00%	-24.88%	-24.47%

(a) Expected improvement in joint reward

% Results Included	100%			Top 25%			Top 5%		
# Smoothing	0	10	100	0	10	100	0	10	100
R/T-DOrg	34.08%	33.02%	33.12%	34.81%	34.04%	33.77%	27.88%	27.79%	27.76%
unfactoredOrg	34.08%	33.00%	33.13%	34.84%	30.83%	30.90%	28.03%	28.63%	29.36%
non-R/T-DOrg	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

(b) Expected reduction in number of states in agent planning process

Table 1: Percent improvement compared to **localOrg** for experiments in Section 4. %-results-included refers to filtering out episodes based upon the magnitude of the percent reward difference until only the top x% of episodes remain (those same episodes are used for the corresponding expected-number-of-states results). #-smoothing refers to the number of times the smoothing filter was applied to each cell in the ODPs input model.

local planning overhead of the agents. A high-performing organization is one that improves joint reward while also simplifying each agent's local planning problem.

To run our experiments, each agent independently overlays its organizational specification (see Section 3) to create its combined local MDP. It then uses this model to create the reachable state space from the episode's initial state forward. The agent creates its optimal local policy for the reachable state space using CPLEX [7] to solve the linear program as formulated by Kallenberg [8]. To reduce the impact of stochastic transitions, we simulated each joint policy 5000 times in the execution environment to calculate the expected joint reward. As we did previously [11], if during execution an agent reaches an unplanned state (e.g., due to another agent unexpetedly fighting a fire), it constructs a new policy going forward from its current state.

Table 1 presents results from our experiments and highlights several important points. Firstly, we observe that, in expectation, the value of organizational influence is only 1.45% in this domain. While this initially seems disappointing, on reflection it is unsurprising: for most episodes, an agent's lack of global awareness is inconsequential, because the initial placement of agents (Figure 1) and fires is such that, for most cases, agents' local decisions lead them into complementary actions. However, in episodes where a high-intensity fire is located between the agents' initial positions, agents acting locally often miscoordinate, providing opportunities for organization to improve performance. This suggests that the average performance hides a heavy-tailed distribution. If we sort the episode results by the magnitude of the percent difference versus **localOrg**, $\frac{|localOrg-org|}{localOrg}$, and filter the sorted results to only include the top percentages, then we observe that when organizational design does impact performance, it has a noticeable impact. For example, **R/T-DOrg** has a 23.27% expected improvement in 5% of the episodes. Note that in Table 1b, the improvement declines of **unfactoredOrg** and **R/T-DOrg** from the 25% to 5% settings are also caused by this domain property. The R/T-D based organizations coordinate correctly when a high-intensity fire is located between the agents due to I_c transition shaping; however, transition shaping also increases the size of the agents' state spaces.

Secondly, as can be seen in any of the columns, the organizations that influence R/T-D factors (i.e., **unfactoredOrg** and **R/T-DOrg**) outperform those without R/T-D based influences, both in terms of expected joint reward as well as computational costs. Moreover, the non-R/T-D based influences can severely degrade system performance as the organization's model deviates from the agents' true models, as demonstrated by the 10- and 100-smoothed cases for **non-R/T-DOrg** and **unfactoredOrg**. This illustrates the costs of heavy-handed micromanagement that undervalue agents' expertise, and supports our claim of superiority for our factored organizational formulation, as organizations that omit non-R/T-D based influences allow agents to exercise their expertise to avoid these problems.

Unfortunately, we are unable to fully contrast our organizations against computing the optimal joint policy, π^*, as we were computationally unable to calculate π^* for all 3000 experiments. However, in a small subsample of the problems that we could complete, we observed that π^* contains approximately two orders of magnitude more states and achieves approximately three percent more expected reward as compared to the **localOrg**.

5. AUTOMATED DESIGN

Intuitively, organizational design should use a global perspective to identify patterns of interactions that would arise when agents cooperate effectively, and then codify these patterns into influences that agents internalize. In Section 4, for example, the **R/T-DOrg** stops agents from even thinking about fighting fires that another agent is clearly better positioned to fight, and focuses them on fighting nearby fires. Furthermore, the factored model and organizational design principle we presented in the previous sections suggest the foundations of a process for automating organizational design: identify the R/T-Ds using the Dec-POMDP model; deduce from these a space of joint behaviors to seek/avoid; and use these to select and shape factors for agents' local models that steer agents to/from local decisions that lead to the good/bad interactions.

Unfortunately, identifying R/T-Ds can sometimes be difficult in models where joint interactions are not explicitly provided as part of the model specification. For this rea-

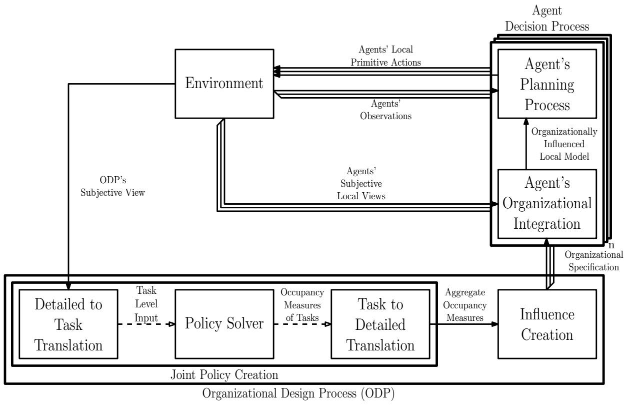


Figure 4: A conceptual overview of our ODP and how it interacts with the environment and the agents.

son, the automated organizational design process (ODP) we have devised exploits domain knowledge if it is provided, but still functions (although with increased computational costs) without explicit R/T-D specifications. At a high level (Figure 4), our ODP begins with joint policy creation, wherein it calculates optimally-coordinated joint policies over a space of possible problem instances to estimate the aggregate occupancy measures for the problem space. An occupancy measure $x(s, a)$ for a policy tree gives the probability of reaching state s and taking action a, and is directly optimized via the linear program [8] our ODP uses for policy creation. Our ODP then uses the aggregate occupancy measures as the basis for influence creation, such that Θ's influences guide the agents into the behavior patterns captured by the aggregate occupancy measures.

5.1 Organizational Design Process

We now step through our currently implemented automated ODP in more detail, and then at the end of this section discuss some ways in which it can generalize to incorporate additional knowledge if it is available. We assume there is a "true" environment in which the agents are operating, but neither an agent nor the ODP is assumed to have a perfect model of that environment. Rather, each agent has a subjective local view, which is represented as a local MDP, such as those described in Section 2, where an agent does not model other agents but can have accurate models of the environment such as the delays (as in the experiments in Section 4). The ODP has a subjective view of the environment and agents in the form of a Dec-POMDP, describing the environment as well as the agents and their capabilities, but may be imperfect in either or both aspects (e.g., imprecise models of the cell delays as in Section 4).

Our ODP begins by using the options framework [13] from the hierarchical learning community to abstract its Dec-POMDP into a task-level model that focuses on tasks to accomplish rather than actions to take. As is customary, one option is created for each task in the domain, where a task corresponds to achieving a particular subgoal. For simplicity, in the experiments that follow, we informed our ODP that good subgoals are states where $I_c \to 0$; however, subgoal detection could be automated using techniques from the hierarchical learning community (e.g., [12]). Reasoning with task-level options not only reduces computation, but also naturally emphasizes the most significant interactions among agents, while remaining largely agnostic about how the agents will translate their options into detailed actions. Of course, the ODP still requires an estimate of each option's properties (i.e., its expected reward, termination states, and primitive actions it might translate into). Our ODP creates this information itself by using its detailed Dec-POMDP to heuristically calculate the likely effects of an agent's policy within an option.

Our ODP then solves a linear program representation of the task-level Dec-POMDP (as a centralized process) analogously to the policy-solving process our agents used in Section 4. This results in occupancy measures for state-option pairs in the joint task-level policy. The ODP then inverts its abstraction process using the properties of each option, which projects state-option occupancy measures downward to estimate state-action occupancy measures. This joint policy creation process (i.e., abstracting to a task-level model, solving for the task-level joint policy, then inverting the abstraction) is repeated for a space of problems sampled from the Dec-POMDP, which results in aggregate occupancy measures that identify patterns of optimal task-level interactions.

Specifically, our ODP uses the aggregate state-action occupancy measures to create influences for the agents from the following patterns.

Actions: For agent i, if the occupancy measure $x(s_i, a_i) = 0$ then block a_i from the set of available actions in s_i. For example, in the firefighting domain, if the ODP's policies never have an agent move into certain cells (e.g., cells always serviced by closer agents), then actions that would move the agent into those cells are blocked.

States: If agent i's action choice under the joint policy is invariant with respect to state factor F_{i_k} given any values for the other state factors, then F_{i_k} can be blocked from agent i's state factors (since it contributes no information). For example, in the firefighting domain, the intensity of cells distant to an agent (always fought by someone else) do not impact the agent's action selection and thus are blocked.

Transitions: For an agent, modify the transition factors for each of its remaining state factors (after blocking state factors as above) to include the probabilistic effects of the other agents. For example, in the firefighting domain, an agent's transition factor for an overlapping cell's intensity would be altered to reflect the probability that some other agent executes the FF action in that cell at certain times.

Notice that the above influence mechanisms use very strict criteria for when to remove a factor. Essentially, our ODP finds the maximal reduction to an agent's model that does not decrease the expected joint reward. In principle, however, further reductions could sacrifice reward in order to further influence the agents. For example in the firefighting domain, it could be the case that agent i rarely needs to know the intensity of a relatively distant cell, so the expected reward loss from not knowing it is very small. Thus, blocking that factor has negligible impact on the expected joint reward, but could yield significant computational savings during agent planning. Tradeoffs like these are the focus of model abstraction research [9], and in the future we plan to extend our work to address these issues.

We now briefly discuss ways of generalizing our ODP to incorporate additional knowledge, should it be available. If an explicit specification of the R/T-Ds is provided, then the joint policy creation process could simply create a policy directly from the R/T-Ds. Further, if (partial) information of good joint interactions is known (e.g., a partial-order plan), then the joint policy creation process could be constrained to reflect that knowledge. Alternatively, knowledge of the R/T-Ds could be reflected by simply inputting a corresponding option-level model to the ODP (along with properties of the options to be used for inverting the abstraction) as opposed to a detailed Dec-POMDP. Moreover, if the ODP has an option-level model, but lacks knowledge of how options translate into local actions, states, and transitions, it could influence agents by adding/blocking local reward factors to induce coordinated behavior. Reward influences thus provide a fallback means for exerting organizational influences. Typically, however, action, state, and/or transition influences are preferable since they can reduce agents' reasoning efforts by preventing consideration of ineffectual behaviors.

5.2 Evaluation

Before presenting our evaluation, we must first discuss the parameters of our automated ODP in these experiments. That is, even in the relatively simple firefighting domain with restrictions on the starting positions of the agents and a maximum number of fires and their intensities, the space of possible initial global states is exceedingly large (∼22,000). Furthermore, the total reachable state space from any initial state contains millions of states. For these reasons, our ODP is computationally unable to exactly solve for the complete, optimal joint policy in every state. Thus our ODP instead bases its designs on a representative sub-sampling from the entire initial state distribution (i.e., we biased the sampling to ensure that the ODP samples a diverse set of initial states). To test the impact the sample size has on the resulting organizations, we present results from two different parameter settings, 50 (0.23% of possible initial states) and 150 (0.68%). ***X*AutoOrg** refers to the organization designed by our ODP using $X \in \{50, 150\}$ initial state samples.

We begin our evaluation by confirming that our ODP's designs are intuitively sensible. Figures 5a/5b show the cumulative occupancy measures by cell (shaded by magnitude), $\sum x_i(s_i, \cdot)$, for each agent, created in response to the delays in Figure 5c, and represent a summary of the action shaping specified to each agent (i.e., cells with low cumulative occupancy measure typically have more tightly restricted actions). Darker shaded cells thus represent those that the ODP expects the agent will more likely visit. We observe in Figures 5a and 5b that the agents' influences are correlated, and each agent is more or less expected to be responsible for a particular region (with some overlap in between). Further, as seen by comparing Figure 5c to Figures 5a/5b, the ODP recognized the domain structure, and tailored its influences to skew the agents' regions towards those cells they can easily reach.

We also tested the ***X*AutoOrg**s using the same empirical methodology and episodes as in Section 4 to ensure that they yield high system performance in addition to being intuitively sensible. Table 2 presents the results of these experiments as well as repeats the results from the best hand-designed organization from Section 4. We present only the results for the 0-smoothed case; however, the other cases are nearly identical to the 0-smoothed one, implying that our ODP scales gracefully as its input model degrades (due to following our principle of specifying R/T-D based influences).

% Results Included	**100%**	**Top 25%**	**Top 5%**
R/T-DOrg	1.45%	6.05%	23.27%
50AutoOrg	2.06%	3.33%	4.11%
150AutoOrg	2.17%	5.25%	13.63%

(a) Expected improvement in joint reward

% Results Included	**100%**	**Top 25%**	**Top 5%**
R/T-DOrg	37.07%	34.81%	30.39%
50AutoOrg	38.84%	39.29%	49.15%
150AutoOrg	19.36%	20.75%	38.91%

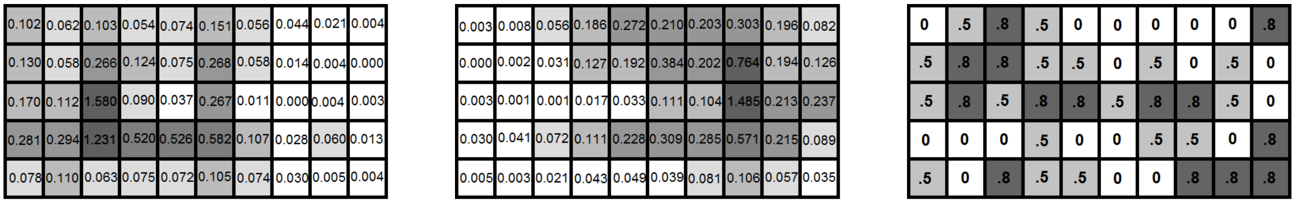
(b) Expected reduction in number of states in agent planning

Table 2: Percent improvement compared to **localOrg** for experiments in Section 5. %-results-included data uses the same subset of episodes as in Table 1.

As Table 2 illustrates, our ***X*AutoOrg**s compare well against the hand-designed **R/T-DOrg**, but make different tradeoffs as demonstrated by the top 25% and 5% columns. That is, **R/T-DOrg** has little to no impact in most episodes, but then has substantial gains in a few episodes, whereas the ***X*AutoOrg**s yield a slightly larger overall improvement, but accomplish this by having moderate performance gains in many episodes. This observation suggests that the ***X*AutoOrg**s are not over-specialized to particular situations the ODP might have encountered, but rather provide general influences, since their improvement gains are more uniformly distributed over the problem space relative to **R/T-DOrg**. We also observe that, as our ODP gains a more complete input model, it uses this additional information to infer more specialized behavior patterns and thus exerts more specialized influences, as evidenced by the **150AutoOrg** having moderately higher performance gains relative to **50AutoOrg** in the 25% and 5% cases. Finally, we observe that the ***X*AutoOrg**s' state space improvements actually increase as we filter out episodes—in stark contrast to the **R/T-DOrg**. Recalling from Section 4, the episodes where organizational influence are most meaningful are those where a high-intensity fire is between the agents' initial locations. While the **R/T-DOrg** approaches these cases with I_c transition shaping, the ***X*AutoOrg**s instead address them with state/action shaping (e.g., by delegating the northern cells to one agent and the southern to the other), which reduces the state space rather than increasing it.

6. RELATED WORK

One body of related work is organizational modeling languages (OMLs) such as MOISE$^+$ [6] and OMNI [15], among many others. OML research generally takes a problem-centric perspective, by creating a formal syntax that represents how to decompose and solve a target problem in terms of organizational roles for handling subproblems, role relationships for addressing subproblem interactions, etc. An organizational designer thus uses expertise about the target multiagent problem to specify a corresponding organization via an OML. The automated organizational design processes of Horling [5] and Sims [10] extend the OML work by searching for an optimal decomposition strategy, as configured from a library of problem-appropriate organizational goals, constraints, roles, agent capabilities, etc. Problem-centric approaches are particularly suited to open agent systems, where the organizational structure is built to address long-term problem needs, and



0.102	0.062	0.103	0.054	0.074	0.151	0.056	0.044	0.021	0.004
0.130	0.058	0.266	0.124	0.075	0.268	0.058	0.014	0.004	0.000
0.170	0.112	1.580	0.090	0.037	0.267	0.011	0.000	0.004	0.003
0.281	0.294	1.231	0.520	0.526	0.582	0.107	0.028	0.060	0.013
0.078	0.110	0.063	0.075	0.072	0.105	0.074	0.030	0.005	0.004

(a) Agent 1

0.003	0.008	0.056	0.186	0.272	0.210	0.203	0.303	0.196	0.082
0.000	0.002	0.031	0.127	0.192	0.384	0.202	0.764	0.194	0.126
0.003	0.001	0.001	0.017	0.033	0.111	0.104	1.485	0.213	0.237
0.030	0.041	0.072	0.111	0.228	0.309	0.285	0.571	0.215	0.089
0.005	0.003	0.021	0.043	0.049	0.039	0.081	0.106	0.057	0.035

(b) Agent 2

0	.5	.8	.5	0	0	0	0	0	.8
.5	.8	.8	.5	.5	0	.5	0	.5	0
.5	.8	.5	.8	.8	.5	.8	.8	.5	0
0	0	0	.5	0	0	.5	.5	0	.8
.5	0	.8	.5	.5	0	0	.8	.8	.8

(c) Cell Delays δ_c

Figure 5: Cumulative cell occupancy measures, $\sum x_i(s_i, \cdot)$, for each agent that our ODP calculated in response to the δ_c's in (c)

the organization persists even though agents performing particular roles can change.

In contrast to a problem-centric perspective, our work is agent-centric. We assume a group of agents in a multiagent system already intends to cooperatively solve problems in their environment, and might even already be working together. The purpose of forming an organization, in this context, is to explicitly reason over and codify expectations about appropriate behaviors and interaction patterns in order to improve and streamline cooperation. Agent-centric approaches are thus advantageous for designing organizations to fit the objectives, capabilities, and limitations of a group of agents that will be cooperating over an extended time, even though the the problems they face might vary. Hence, problem-centric and agent-centric approaches both emphasize the design of stable organizations, but differ in which aspects of the agents' task environment they treat as stable.

7. CONCLUSION

In this paper we presented a formal framework for specifying factored organizational influences and incorporating them into agents' decision models. We then argued that an ODP should restrict itself to R/T-D based influences, and empirically demonstrated the effectiveness of this design principle. Finally, we presented an automated ODP based upon this principle, and demonstrated how it creates sensible specifications that exploit structure within the domain. In the future, we plan to expand upon the functionality of our ODP, empowering it to make informed tradeoffs between restricting agent behaviors and the resulting potential loss of reward (e.g., by utilizing different, more sophisticated influence creation mechanisms). Additionally, we plan to relax some of the simplifying assumptions throughout this paper such as requiring that each agent always fully adopt its θ_i, for example, with organizationally adept agents [3].

8. 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.

9. REFERENCES

[1] M. Babes, E. M. de Cote, and M. L. Littman. Social reward shaping in the prisoner's dilemma. In *AAMAS*, pages 1389–1392, 2008.

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

[3] D. D. Corkill, C. Zhang, B. da Silva, Y. Kim, X. Zhang, and V. R. Lesser. Using annotated guidelines to influence the behavior of organizationally adept agents. In *COINS2012 Workshop at AAMAS*, 2012.

[4] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman. Efficient solution algorithms for factored MDPs. *JAIR*, 19:399–468, 2003.

[5] B. Horling and V. Lesser. Using quantitative models to search for appropriate organizational designs. *JAAMAS*, 16(2):95–149, 2008.

[6] J. Hubner, J. Sichman, and O. Boissier. Developing organised multiagent systems using the MOISE$^+$ model: programming issues at the system and agent levels. *IJAOSE*, 1(3):370–395, 2007.

[7] IBM. Ibm ilog cplex, 2012. See http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

[8] L. C. M. Kallenberg. *Linear Programming and Finite Markovian Control*. Mathematical Centre Tracts, 1983.

[9] L. Li, T. J. Walsh, and M. L. Littman. Towards a unified theory of state abstraction for MDPs. In *ISAIM*, pages 531–539, 2006.

[10] M. Sims, D. Corkill, and V. Lesser. Automated organization design for multi-agent systems. *JAAMAS*, 16(2):151–185, 2008.

[11] J. Sleight and E. H. Durfee. A decision-theoretic characterization of organizational influences. In *AAMAS*, pages 323–330, 2012.

[12] M. Stolle and D. Precup. Learning options in reinforcement learning. In *Lecture Notes in Computer Science*, pages 212–223, 2002.

[13] R. S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *AI*, 112:181–211, 1999.

[14] P. Varakantham, J. Kwak, M. Taylor, J. Marecki, P. Scerri, and M. Tambe. Exploiting coordination locales in distributed POMDPs via social model shaping. In *ICAPS*, pages 313–320, 2009.

[15] J. Vázquez-Salceda, V. Dignum, and F. Dignum. Organizing multiagent systems. *JAAMAS*, 11:307–360, 2005.

[16] P. Velagapudi, P. Varakantham, K. Sycara, and P. Scerri. Distributed model shaping for scaling to decentralized POMDPs with hundreds of agents. In *AAMAS*, pages 955–962, 2011.