Effective Influence Abstractions for Organizational Design

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

ABSTRACT

Organizational structures often employ abstraction to specify broadly-applicable behavioral roles and guidelines to agents, where abstraction streamlines the design process and leaves agents room for tailoring actions to the evolving situation. Too much abstraction, however, can provide too little guidance towards fruitful joint activities or burden agents with solving complex coordination problems. In this paper, we examine how abstraction choices for organizational influences affect both the process of designing organizations and the performance of a multiagent system using designed organizations. To do so, we identify dimensions of abstraction pertinent to organization design processes and outcomes. Mapping these dimensions to an example domain, we empirically evaluate organizations at different points in the abstraction space to inform an analytical framework for understanding the impact of abstraction in organizational design. Finally, we use our new framework to converge on task-delineated abstractions as a general-purpose organizing heuristic, and confirm this heuristic's effectiveness empirically.

Categories and Subject Descriptors

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

General Terms

design, measurement, performance, experimentation, theory

Keywords

organizational design; abstraction; multiagent coordination; multiagent decision making; metareasoning

1. INTRODUCTION

Organizational structures for multiagent systems are often composed of influences on agent behavior, such as incentives to reward cooperation, or restrictions on action choices to prevent conflicting behaviors. Abstraction plays a central role in specifying such influences. For example, broad, encompassing influences can be advantageous if an organization wants to instill overarching organizational guidance for the agents' operational decisions. Alternatively, sets of narrow, detailed influences can be advantageous if an organization wants to ensure specific coordination patterns tailored for particular environmental conditions. This range of desirable abstraction mechanisms emphasizes the importance of carefully selecting an abstraction for organizational influences.

Recent work in organizational modeling languages and automated organizational design processes (ODPs) has recognized the significance of abstraction choice for creating effective organizational designs [4, 5, 6, 11, 12, 13], and typically includes mechanisms for an organizational designer to formulate abstract influences. However, the field still lacks a systematic understanding of how abstraction choices impact organizational design processes and outcomes. As the first main contribution of this paper, we enumerate pertinent dimensions of abstract organizational influences, and empirically evaluate families of organizations that we hand-crafted to represent several points along these dimensions. We then use our empirical findings to develop an analytical framework for understanding how abstraction dimensions impact an ODP as well as the operational characteristics of multiagent systems using the organizations the ODP designs.

For our second main contribution, we use our analytical framework to derive a general-purpose heuristic for selecting abstract organizational influences based on task-delineated phases of the agents' decision problems. Our empirical results demonstrate that this approach has comparable operational performance to our best hand-crafted abstraction, but is more robust to inaccurate statistical estimates in the ODP.

We begin in Section 2 by discussing prior approaches from the literature and their relation to abstraction in organizational designs. Then, in Section 3 we present an example problem domain that we use for illustration and evaluation throughout the paper, and describe the specific ODP approach that we extend to accommodate abstract influences. Building on this foundation, we precisely define what abstract organizational influences mean in this context, and enumerate the pertinent dimensions for abstraction in organizations (Section 4). In Section 5, we use these dimensions to construct a space of abstraction approaches, and empirically evaluate their effectiveness. These results form the foundation for our analytical framework for understanding the impact of abstraction dimensions on organizational design processes and outcomes. Using our framework in Section 6, we derive task-delineated abstractions as a general-purpose organizing heuristic, and empirically demonstrate their effectiveness. Finally, we conclude in Section 7 with a summary of our findings and some directions for further investigation.

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May, 4–8, 2015, Istanbul, Turkey.*

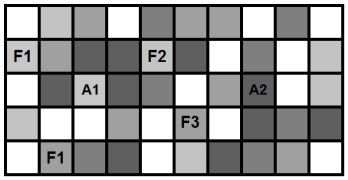


Figure 1: Example initial state in our firefighting grid world. Ai is the position of agent i, and Fx indicates a fire with intensity x. Darker shading indicates higher delay.

2. RELATED WORK

In automated ODPs such as [5, 11, 12, 13], the ODP's organizational design space imposes the degree to which organizational influences will be abstracted, where for example a design space with detailed specifications implies a narrow abstraction and *vice versa*. While each of these prior ODPs has inherently used an abstraction, little explicit discussion has been devoted to the question of how to choose an appropriate abstraction. For example, in prior work [13], we noted that abstraction choice can have substantial impact on the effectiveness of an ODP; however, we did not perform an in-depth investigation of how abstraction choice affects the ODP algorithm or the performance of the organizations it designs. This paper serves to fill that gap, and provides insights for selecting an appropriate abstraction to use in conjunction with automated ODPs.

In order to provide expressive flexibility for a spectrum of system requirements, recent organizational modeling languages often explicitly distinguish between various abstractions that an organization can utilize. For example, OMNI [4] specifies organizations at abstract (e.g., main organizational objectives), concrete (e.g., social and interaction structures), and implementation (e.g., norm interpretation) levels, and MOISE$^+$ [6] specifications include functional (e.g., schemes for goal achievement), structural (e.g., roles and their relationships), and deontic (e.g., duties associated with roles) dimensions. However, beyond high-level intuitions such as the modeling semantics associated with each aspect of the language, there has been insufficient research on when it is appropriate to utilize a particular construct of a language, or on how different organizational encoding approaches will affect the organizational design process and the resulting organization's performance. By providing a deeper investigation of how abstraction choices impact organizations, our work here provides complementary knowledge for guiding usage of these languages.

In multiagent sequential decision making research, several approaches exist for incorporating abstractions into the agents' representations and reasoning processes, although the approaches view abstraction from the operational decision making perspective as opposed to an organizational design perspective. For example, research into finite state controllers [3, 8] investigates abstracting histories to create policies of bounded, finite size. Influence decoupling frameworks [16] examine how multiagent coordination can be efficiently facilitated through influence abstractions that focus how the agents' can affect each others' policies. Coordination locale frameworks [15] study how multiagent coordination can be greatly simplified if the coordination possibilities can be predetermined and localized. Options research [1, 14] shows how agents' behaviors can be abstracted and viewed as macro-actions that can speed up policy creation. Each of these approaches (among others not mentioned here) explores various tradeoffs between the computational efficiency gained from abstract reasoning or representations versus the (potential) loss of resulting solution quality, and we leverage these ideas to guide and inform our investigation of abstractions as related to organizational influence specifications.

3. BACKGROUND

3.1 Example Domain

For the remainder of this paper, we adopt our previously developed organizational design problem [13]. We assume the system is represented as a locally-fully observable decentralized Markov decision process (Dec-MDP) [2] in which each of n fully-cooperative agents possesses a local model, $\mathcal{M}_i = \langle S_i, \alpha_i, A_i, P_i, R_i, T_i \rangle$, that specifies agent i's: factored state space (S_i), initial state distribution (α_i), action space (A_i), transition function (P_i), reward function (R_i), and finite time horizon (T_i). Using any of the standard techniques [9], each agent can calculate an optimal local policy, $\pi_i^* : S_i \times A_i \mapsto [0, 1]$ from its $\mathcal{M}_i$. These π_i^*s together constitute the joint policy for the system, $\pi = \langle \pi_1^*, \cdots, \pi_n^* \rangle$, which may or may not be jointly optimal depending on how the agents coordinate the construction of their local policies. The joint Q-value, $Q^\pi(s, a)$, is defined as the expected cumulative reward of executing joint action a in global state s, then following joint policy π.

To illustrate a problem of this type, we previously developed a simplified firefighting domain. To summarize the domain (see [13] for a more complete description), two agents move through a grid world to fight fires of various intensities until some predetermined, fixed time horizon. Each cell has a delay that stochastically prevents movement into that cell (e.g., rubble or traffic), and the joint reward is determined by the intensities of all the active fires in the grid. Figure 1 shows an example initial state with four fires, though in [13] there was a maximum of two fires.

It is important to recognize that while the firefighting domain may seem straightforward from the perspective of the agents' decision problems (despite there typically being millions of states in the joint problem), the complexity of the agents' problems is of secondary importance to the complexity of the organizational design problem. In this respect, the firefighting domain presents several interesting challenges. For example, since the local behaviors of the unorganized system naturally perform quite well (even optimally in some cases), an ODP must create an exceptional organization in order to outperform the local baseline. Additionally, the range of optimal coordination patterns is diverse (e.g., agent 1 should fight all of the fires in some random instances, only one fire in others, etc.), which compels an ODP to create flexible, robust organizations.

3.2 Organizational Design Problem

We build upon our previous work [13] and define an **organizational influence**, $\Delta_i : S_i \times A_i \times S_i \mapsto S_i \times A_i \times S_i$ as a modification to $\mathcal{M}_i$ at $(s_i \in S_i) \times (a_i \in A_i) \times (s_i' \in S_i)$. Alternatively, it is sometimes useful to view Δ_i as a constraint or re-prioritization to the agent's local policy space brought about by a modification to $\mathcal{M}_i$. For example in the firefighting problem in Figure 1, a Δ_i could alter agent 1's $\mathcal{M}_i$ to not include the move-east action in the initial state, which

prevents agent 1 from selecting polices with this state-action mapping and biases it towards fighting the western fires.

An **organizational design**, Θ, is defined as a set of organizational influences for each agent, $\Theta \equiv \langle \theta_1, \cdots, \theta_n \rangle$, where $\theta_i \equiv \{\Delta_i\}$ is the set of organizational influences for agent i. To measure the performance of an organizational design, we utilize metrics for the expected reward, $\mathbb{R}_{Op}(\Theta)$ and computational costs, $\mathbb{C}_{Op}(\Theta)$, of the the agents' joint policy with respect to the organization, $\pi^{|\Theta}$, and define an **operational performance** function, $\mathbb{P}_{Op}(\Theta)$, to select from the Pareto front of these metrics.[1] Below, $C(\pi_i^{*|\theta_i})$ designates agent i's expected local costs to compute $\pi_i^{*|\theta_i}$.

$$\mathbb{R}_{Op}(\Theta) \equiv \sum_{s \in S} \alpha(s) \sum_{a \in A} \pi^{|\Theta}(s,a) Q^{\pi^{|\Theta}}(s,a)$$
$$\mathbb{C}_{Op}(\Theta) \equiv E_i[C(\pi_i^{*|\theta_i})]$$
$$\mathbb{P}_{Op}(\Theta) \equiv f(\mathbb{R}_{Op}(\Theta), \mathbb{C}_{Op}(\Theta))$$

Returning to the firefighting example, by carefully selecting appropriate influences like the one discussed above, a Θ can designate regions that each agent is responsible for fighting fires within (i.e., remove actions that if left available would permit an agent to leave its organizationally designated region). Such an organization can decrease the amount of computation that each agent requires to find its policy (i.e., $\mathbb{C}_{Op}(\Theta)$), since an agent is not permitted to consider actions that would take it out of its region. In exchange, the expected joint reward (i.e., $\mathbb{R}_{Op}(\Theta)$), might decrease if there are specific problem instances where an agent should leave its region (e.g., if there are no fires in its region).

The purpose of an ODP is to find a Θ that optimally balances these objectives in order to maximize the operational performance, $\Theta^* \equiv \text{argmax}_\Theta \, \mathbb{P}_{Op}(\Theta)$. Unsurprisingly, finding such an organization is computationally intractable, thus we developed an efficient algorithm for finding a locally optimal Θ [13]. To summarize, our algorithm computes the incremental impact of an individual Δ_i with respect to a candidate organization, then embeds these calculations within a greedy hill climbing algorithm. Abusing notation, the algorithm in standard linear approximation form is:

$$\Theta^{j+1} = \Theta^j + \underset{\Delta_i}{\text{argmax}} \left[\Delta_i \cdot \frac{d\mathbb{R}_{Op}}{d\Theta^j}(\Theta^j) \frac{\delta f}{\delta \mathbb{R}_{Op}}(\Theta^j) + \Delta_i \cdot \frac{d\mathbb{C}_{Op}}{d\Theta^j}(\Theta^j) \frac{\delta f}{\delta \mathbb{C}_{Op}}(\Theta^j) \right]$$

A main contribution of this prior work was a set of techniques for efficiently computing the $\Delta_i \cdot \frac{d\mathbb{R}_{Op}}{d\Theta^j}(\Theta^j)$ and $\Delta_i \cdot \frac{d\mathbb{C}_{Op}}{d\Theta^j}(\Theta^j)$ terms of this algorithm. To summarize those techniques, the ODP samples instances from a global perspective of the domain and computes optimal joint policies for each sample, then aggregates across these samples to calculate the expected impact of an influence.

4. ABSTRACT ORGANIZATIONAL INFLUENCES

In the work just summarized, the ODP utilized an abstraction that relied on only an agent's position to identify patterns in that agent's reasoning and behaviors. The resulting organizations exploited specific problem properties—especially that a problem instance often had exactly one fire for each agent to fight—and specified influences that essentially captured that, once an agent starts moving toward a fire, it shouldn't think about reversing its movements, and that it is not useful to (think about) moving to places more easily reached by the other agent. The ODP discovered that, because each agent typically fought only one fire, an organization that disallows reverse movements and movements deep in to the other agent's territory performs well.

To pose greater challenges to the system and ODP, we make two extensions to that firefighting domain. First, we extend the number of fires per problem to four. Second, in previous versions of the domain, there were implicitly minimum durations before the agents could impact each other (i.e., determined by the minimum number of moves required to reach grid cells with active fires), but due to the cell delays there was no finite maximum duration (other than the time horizon). To provide meaningful finite maximum durations before an agent could impact another, we introduce a cap on the maximum number of consecutive failed movement attempts before success is ensured. We maintain the Markov property by adding a new state factor to maintain the number of consecutive failed moves.

Since each agent might fight multiple fires in our extended version of the domain, an encompassing abstraction that at all times prohibits reverse movements could stop the agents from reaching a second fire, and as a result have poor operational performance. Intuitively, however, a narrower abstraction (e.g., that partitions time) could provide the necessary organizational expressivity for the ODP to differentiate these coordination patterns in a more nuanced specification.

4.1 Motivation for Abstract Influences

We have two primary motivations for abstract organizational influences in this paper:

1) By generalizing where influences apply beyond just the seen instances, an appropriate abstraction can improve organizational performance. For example in the firefighting domain, generalizing instances of purposeful movement toward a fire to prohibit reverse movements everywhere.

2) By abstracting over a wider space of instances, an appropriate abstraction can find influence patterns with greater confidence (i.e., it avoids overfitting). For example in the firefighting domain, seeing enough instances to confidently constrain the agents to local partitions of the grid world.

Of course, these benefits can be lost if abstraction is taken too far. Overextending abstraction can misapply influences, and conflate patterns or properties of influences that can harm operational performance and/or confuse an ODP's search algorithm. Alternatively, too little abstraction can too sparsely distribute the ODP's limited information, yielding poor statistical estimates that make the ODP's search algorithm sensitive to sampling artifacts, and organizational performance reflective of the agents' arbitrary priors.

Beyond these fundamental motivations, abstract influences could have several other benefits depending on the application. One intuitive example is that abstract organizational influences can reduce organizational specification size. This can be important when an agent queries its θ_i to find the influences associated with the part of its $\mathcal{M}_i$ currently being considered. In our computational agents, however, we employ hashing to provide $O(1)$ query complexity for organizational lookup, which makes specification size a non-issue.

[1] We assume the Pareto optimality function f is monotonic in each dimension s.t. higher $\mathbb{R}_{Op}$ and lower $\mathbb{C}_{Op}$ is preferable.

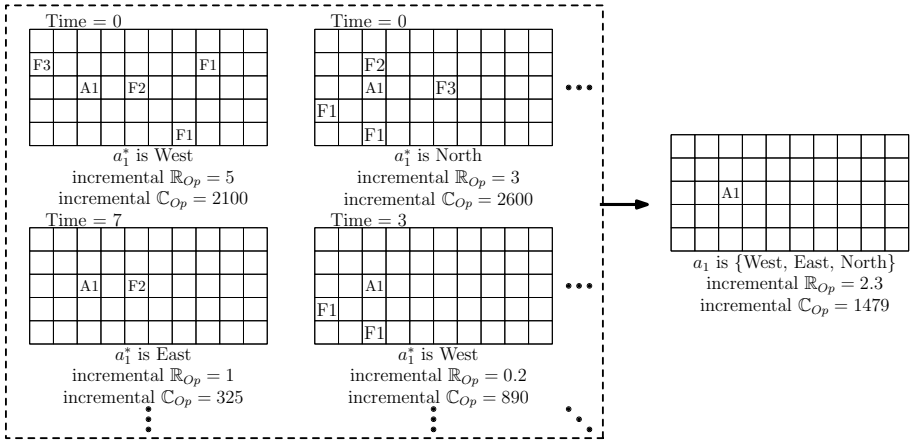


Figure 2: Illustration of the Pos abstraction's dimensions (see Table 1) for agent 1's initial position. The influence's inclusivity is $\frac{1}{50}$, its uniformity is $\frac{1}{3}$, and its $(\mathbb{R}_{Op}, \mathbb{C}_{Op})$ variance is $(4.63, 1.11 \times 10^6)$. Quantities shown are for the four example Δ_1s (rather than $\hat{\Delta}_1$'s entire domain), and are for illustrative purposes only and not from empirical data.

The ODP's computational costs are another possible motivating factor for abstraction, where broader influences imply a smaller organizational design space. The search algorithm from Section 3.2 has complexity $O(|\{\Delta_i\}|^2)$; however, it is important to note that the search's computational costs are an insignificant portion (i.e., $\ll 0.1\%$) of the ODP's total computational costs in our experiments. The bulk of the computation is sampling problem instances and computing optimal joint policies to estimate the influences' incremental impacts, which is not affected by abstraction choice.

Yet another possible motivation for organizational abstraction is to provide flexibility for the agents to make local decisions within, while still providing organizational guidance for that reasoning. However, counter to possible intuitions, abstract organizational influences are orthogonal to the flexibility an agent retains in an organization. For example, highly flexible organizations can be specified as an aggregation of many fine-grained influences (e.g., in s_i consider a_i^1, and in s_i also consider a_i^2, and in s_i also consider a_i^3, etc.). Thus, flexibility stems from both number of influences and their abstraction. Of course, abstraction choice could impact the ODP's decision of which/how-many influences to specify, but such flexibility differences arise from ODP decisions rather than as necessary consequences of the abstraction choice. In future work, we plan to more fully investigate the relationship between abstraction and flexibility.

4.2 Dimensions of Abstract Influences

To construct a framework for analyzing abstraction in organizational designs, in this section we provide more precise definitions for abstract influences and identify dimensions of abstraction pertinent to organizational design and outcomes.

Broadly speaking, an abstract organizational influence clusters together detailed influences and forces the ODP and agents into a monolithic treatment of the clustered influences. Figure 2 illustrates this concept. On the left are examples of optimal actions for A1 to take in its initial position, but sampled for different times and fire configurations. On the right is the abstract influence that, based on the patterns seen in the samples, indicates that at this position (at any time or fire configuration) A1 should just consider any of the West, East, or North movement actions.

Formally, an influence abstraction is a function $G : \Delta \mapsto \hat{\Delta}$, where $\hat{\Delta}_i \in \hat{\Delta}$ is an **abstract organizational influence**. Building from the clustering perspective, we identified three primary dimensions for characterizing abstract influences.

Definition 1. The **inclusivity** of an abstract influence, $\hat{\Delta}_i$, corresponds to how encompassing the influence is. Formally, the inclusivity of $\hat{\Delta}_i$ is the expected fraction of agent i's local model, $s_i \times a_i \times s_i' \in S_i \times A_i \times S_i$, that $\hat{\Delta}_i$ modifies.[2]

Definition 2. The **uniformity** of an abstract influence, $\hat{\Delta}_i$, corresponds to how well its composite influences agree on the local model's modification. Formally, the uniformity of $\hat{\Delta}_i$ is the expected fraction of its composite influences that modify agent i's local model in the same way.

Definition 3. The **variance** of an abstract influence, $\hat{\Delta}_i$, is the expected variance of its composite influences' incremental $(\mathbb{R}_{Op}, \mathbb{C}_{Op})$ impact. This differs from uniformity in that $\hat{\Delta}_i$'s composite influences could all modify $\mathcal{M}_i$ in the same way (have uniformity of 1), but have varied estimates for how meaningful the modification is (thus have high variance).

Figure 2 illustrates how the abstraction's values along each of these dimensions are calculated using an abstraction that drops all state factors except the agent's current position. For each of these dimensions, we define it for an organizational design as the expected value over all of the organization's influences; for example, the inclusivity of Θ is the expected inclusivity of its influences.

4.3 Incorporating Abstract Influences

Incorporating abstract organizational influences into the ODP algorithm from Section 3.2 requires extending the calculation of an influence's incremental impact to an abstract influence's incremental impact. We do this by taking the

[2]Beyond abstracting the domain of influences, one could also envision abstracting the range (i.e., the effect of the modification). However, such an approach often decreases uniformity, which is typically undesirable (see Section 5).

expectation over $\hat{\Delta}_i$'s constituent influences.

$$\hat{\Delta}_i \cdot \frac{d\mathbb{R}_{Op}}{d\Theta^j}(\Theta^j) = E_{\Delta_i \mapsto \hat{\Delta}_i}[\Delta_i \cdot \frac{d\mathbb{R}_{Op}}{d\Theta^j}(\Theta^j)]$$
$$\hat{\Delta}_i \cdot \frac{d\mathbb{C}_{Op}}{d\Theta^j}(\Theta^j) = E_{\Delta_i \mapsto \hat{\Delta}_i}[\Delta_i \cdot \frac{d\mathbb{C}_{Op}}{d\Theta^j}(\Theta^j)]$$

Using Figure 2 as an example, the ODP's estimate for $\hat{\Delta}_i$'s incremental $\mathbb{R}_{Op}$ and $\mathbb{C}_{Op}$ impact is the expected impact of its constituent influences. The ODP search algorithm uses these values exactly as it would with un-abstracted influences.

Agents incorporate abstract organizational influences into their local reasoning in exactly the same way they incorporate un-abstracted influences. As each agent i is solving its local decision problem, it queries θ_i to find the $\hat{\Delta}_i$ that applies to the $s_i \times a_i \times s_i'$ currently being considered within its $\mathcal{M}_i$, and then modifies its $\mathcal{M}_i$ in accordance with that $\hat{\Delta}_i$. For simplicity in this work, we limit our consideration to abstractions where a Δ_i maps to a single $\hat{\Delta}_i$. In other words, G must be many-to-one, which implies the agents will only receive logically consistent Θs that do not entail incompatible modifications for any $s_i \times a_i \times s_i'$. Investigating many-to-many abstractions could be an interesting line for future work, since it provides flexibility for hierarchical organizational influences like those found in modern organizational modeling languages [4, 6] and/or conflicting influences brought about by simultaneous membership in multiple organizations.

For example, as agent 1 is solving for its organizationally optimal policy $\pi_1^{*|\theta_1}$, in any state whose location is its initial position, θ_1 specifies the $\hat{\Delta}_1$ shown in Figure 2. Using $\hat{\Delta}_1$, agent 1 will modify its model to only permit consideration of the West, East, and North movement actions in its currently considered state.

5. INFLUENCE ABSTRACTION EFFECTS

5.1 Methodology

The research community has developed an extensive library of abstraction techniques such as: state abstraction [7] and finite controllers [3, 8] for decision-theoretic problems; influence [16] and coordination locale [15] abstractions for efficient coordination; hierarchical planning [1, 14] and task networks [10] for sequential reasoning; and the various abstract modeling constructs for an organizational modeling language [4, 6], among many others. Two overarching commonalities within these techniques, however, are to approach abstraction as 1) overlooking unimportant or irrelevant information, and/or 2) clustering similar information together. Using these as a basis for designing abstractions over various points along our dimensions, we hand-crafted several families of abstractions for the firefighting domain that are summarized in Table 1. Broadly speaking, organizations created from these abstractions map a set of state factors (for specific mappings see Table 1) to the $\hat{\Delta}_i$ for that state, where $\hat{\Delta}_i$ informs agent i of which actions it should consider for that state. Together, these $\hat{\Delta}_i$s essentially construct regions of responsibility for each agent, which can vary over time if system time is a state factor represented in the abstraction.

We empirically evaluated these abstraction choices using the firefighting domain. To observe the impact of abstract influences with respect to $\mathbb{P}_{Op}$'s Pareto topology (i.e., as agents have more or less available computational resources), we encoded $\mathbb{P}_{Op}(\Theta) = \mathbb{R}_{Op}(\Theta) - \frac{1}{b}\mathbb{C}_{Op}(\Theta)$ for parameter $b >$ 0, and for each abstraction had the ODP create organizations for different values of b. High b values represent when the agents have abundant computational resources relative to the pace of the environment, and *vice versa* for low values.

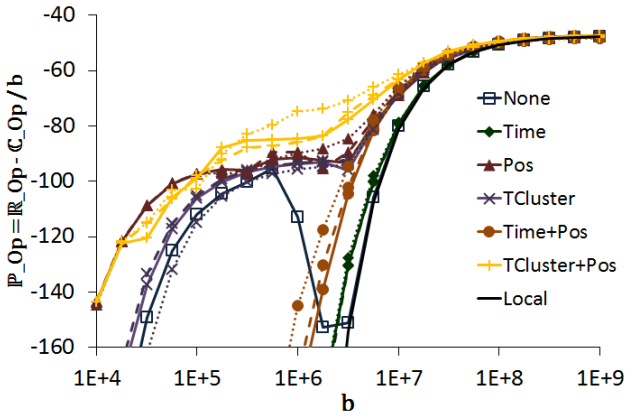


Figure 3: $\mathbb{P}_{Op}$ curves for each abstraction for different amounts of ODP information. Solid, dashed, and dotted curves correspond to perfect information, 2/3 information, and 1/3 information respectively.

To observe the impact of abstract influences with respect the amount of information the ODP possesses, we additionally had the ODP create organizations from three different available information profiles. We controlled the amount of information available to the ODP by artificially manipulating the problem samples from which it constructed estimates of the $\hat{\Delta}_i \cdot \frac{d\mathbb{R}_{Op}}{d\Theta^j}(\Theta^j)$ and $\hat{\Delta}_i \cdot \frac{d\mathbb{C}_{Op}}{d\Theta^j}(\Theta^j)$ terms used in its search algorithm. At one extreme, the ODP exactly sampled the evaluation problem set, which encodes that the ODP has perfect information. Then, as the ODP based its estimates off of fewer sample problems (i.e., the training set is a diminishing subset of the test problems), the ODP possessed increasingly imperfect domain information.

We evaluated each organization on 300 problem instances to empirically compute $\mathbb{R}_{Op}$ and $\mathbb{C}_{Op}$ for each Θ, from which we calculated $\mathbb{P}_{Op}$ for various Pareto optimality parameterizations. Figure 3 shows the $\mathbb{P}_{Op}$ curves for each of these organizations. Figure 4 shows the separate $\mathbb{R}_{Op}$ and $\mathbb{C}_{Op}$ curves for the perfect-information organizations as well as organizations constructed from abstractions we present in Section 6. In the next sections, we systematically analyze these results to develop a framework for characterizing how abstraction dimensions impact: the operational performance of an organization (Section 5.2), the sensitivity of the ODP's search algorithm (Section 5.3), and the effects of information availability on the ODP (Section 5.4).

5.2 Operational Performance

The $\mathbb{P}_{Op}$ curves in Figure 3 reveal two differentiating characteristics for operational performance across abstractions: a curve's smoothness (which will be discussed in Section 5.3) and a curve's raw quality (i.e., its vertical placement on the graph). Analysis reveals that an organization's uniformity is strongly correlated with performance quality. One exemplary case of this is the None abstraction, where the organizations at low b (i.e., when computation is expensive) restrict all but a single action from consideration (i.e., the same action

Abstraction Name	Expected Organizational Inclusivity	Expected Organizational Variance $(10^{-3}\mathbb{R}_{Op}, 10^{3}\mathbb{C}_{Op})$	Description of Θs Constructed from Abstraction
None	1	(2.50, 1297)	Every state maps to the same $\hat{\Delta}_i$.
Time	$\frac{1}{12}$	(35.2, 17.26)	System time in a state maps to $\hat{\Delta}_i$.
Pos	$\frac{1}{50}$	(2.08, 0.99)	Agent's position in a state maps to $\hat{\Delta}_i$.
TCluster	$\frac{1}{4}$	(34.5, 204)	Like **Time**, but system time is clustered into intervals. For example, states with times in $[1, 3]$ map to the same $\hat{\Delta}_i$.
Time + Pos	$\frac{1}{600}$	(18.1, 0.02)	System time and agent's position in a state, together, map to $\hat{\Delta}_i$.
TCluster + Pos	$\frac{1}{200}$	(22.0, 0.22)	Clustered system time and agent's position in a state, together, map to $\hat{\Delta}_i$.

Table 1: Descriptions of the organizational abstractions we evaluate in Section 5. Uniformity for each abstraction relies on the specific Pareto characterization, but typically decreases as agent computation becomes less costly. We evaluated variants of the Pos abstractions using clustered positions, but these abstractions were qualitatively identical to their respective Pos variants.

must be taken in every state). These Θs by definition have maximal uniformity and also obtain relatively high $\mathbb{P}_{Op}$ as compared to other abstractions with similar inclusivity (e.g., Time). Then, as b increases, the None organizations permit consideration of additional actions, eventually decreasing uniformity to its minimal value, and these minimal-uniformity Θs obtain the worst $\mathbb{P}_{Op}$.

Analyzing this observation more deeply, decreased uniformity arises when alternative behaviors could be optimal for specific instances entailed in $\hat{\Delta}_i$'s domain, for example like in Figure 2. Rather than restrict the agent from considering some subset of Pareto-valuable actions that the ODP knows the agent might need in a specific problem instance, the ODP instead permits consideration of all actions that could be Pareto-valuable, and relies on the agent's local intelligence to appropriately select from among this set. As a result, the ODP under-constrains the agents' reasoning, which increases $\mathbb{C}_{Op}$ and thus decreases $\mathbb{P}_{Op}$. However, for high-uniformity influences, the ODP can aggressively restrict the agents' actions to a limited set of Pareto-valuable actions.

$\mathbb{P}_{Op}$'s reliance on inclusivity is interesting in that excess inclusivity can yield poor $\mathbb{P}_{Op}$ (e.g., the None and Time abstractions), and too little inclusivity can also result in poor $\mathbb{P}_{Op}$ (e.g., the Time+Pos abstraction). However, abstractions with moderate inclusivity are associated with the maximal $\mathbb{P}_{Op}$ curves. It is straightforward to show that excess inclusivity increases susceptibility to the effects of decreased uniformity; that is, additional problem situations map into the same $\hat{\Delta}_i$ but may not have the same optimal behaviors. Too little inclusivity is detrimental for the opposite reason; that is, since a low-inclusivity $\hat{\Delta}_i$ applies to such a narrow space, there is insufficient diversity in the ODP's statistical estimates to generalize to unseen problem instances.

Examining the $\mathbb{R}_{Op}$ and $\mathbb{C}_{Op}$ curves in Figure 4 illustrates that, broadly speaking, as the agents' computation becomes relatively cheaper (b increases), the ODP induces the agents to consider more actions, which in turn provides operational flexibility for the agents to achieve higher $\mathbb{R}_{Op}$. The Time and Time+Pos curves deviate from this pattern due to low uniformity across all of the Pareto conditions; that is, the specific behavior an agent should take is poorly correlated with system time, meaning that time-based abstractions cluster together different behaviors. This biases the ODP into permitting the agents to consider additional actions even when Pareto conditions discourage excessive agent reasoning.

5.3 ODP's Search Sensitivity

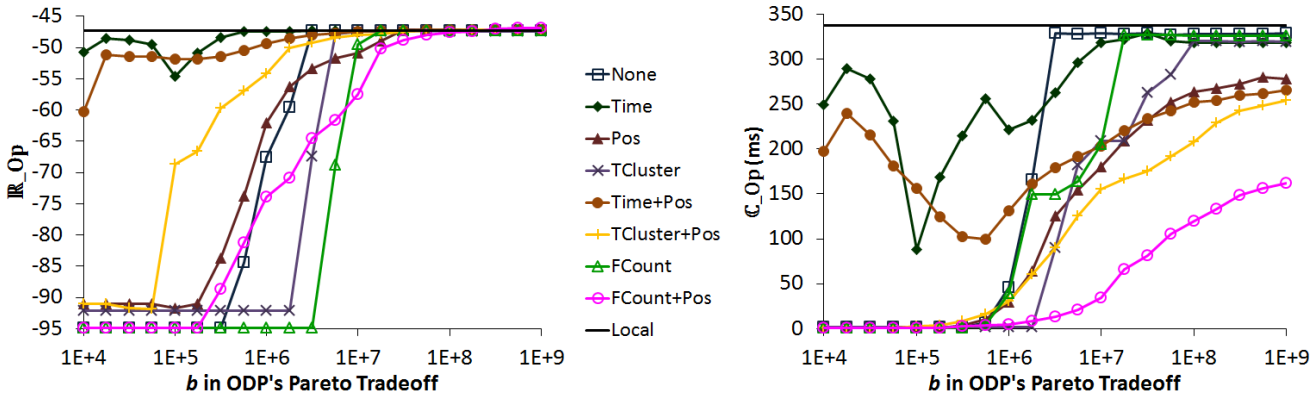
A striking observation from Figure 3 is the large dip in $\mathbb{P}_{Op}$ for some of the abstractions as computation becomes less costly (e.g., in the None abstraction), when we would normally expect smooth, monotonically increasing $\mathbb{P}_{Op}$ curves. In each of these cases, the pre-dip organization would actually be a preferable Θ to the one created by the ODP for these Pareto conditions, which implies that the ODP's greedy search algorithm performed poorly in these instances.

Notice that these dipping cases occur most significantly in the abstractions with high variance. Recalling our definition of the variance dimension, high variance corresponds to $\hat{\Delta}_i$s composed of Δ_is with significantly different expected values for $\Delta_i \cdot \frac{d\mathbb{R}_{Op}}{d\Theta^j}(\Theta^j)$ and $\Delta_i \cdot \frac{d\mathbb{C}_{Op}}{d\Theta^j}(\Theta^j)$, which in turn makes the incremental impact of $\hat{\Delta}_i$ have high statistical variance. Utilizing such imprecise estimates for a $\hat{\Delta}_i$'s incremental impact naturally makes the greedy search algorithm sensitive to initial conditions and small data errors introduced from the ODP's sampling process, which results in the unexpected $\mathbb{P}_{Op}$ performance dips. In other words, the ODP believes from its data that certain influences will improve $\mathbb{P}_{Op}$, when in reality, the ODP is overestimating the reward and/or underestimating the computational costs of adding the influences to the design, and experimentation in the evaluation domain ultimately reveals that these influences are actually detrimental to operational performance.

5.4 ODP's Information Scope

While Figure 4 omits the $\mathbb{R}_{Op}$ and $\mathbb{C}_{Op}$ data for organization's created when the ODP had limited information, the almost unanimous trend is that Θs created from more information have higher $\mathbb{C}_{Op}$ and $\mathbb{R}_{Op}$ than those created from less information. The few exceptions to this trend are caused by search sensitivities like those described in Section 5.3.

Turning to Figure 3, we unexpectedly observe that organizations created from less information tend to achieve higher $\mathbb{P}_{Op}$, whereas intuition would dictate that additional information should tend to improve the quality of an organizational design. Further analysis reveals that an abstraction's inclusivity is a primary determining factor for analyzing an ODP with respect to its available information. As the ODP receives additional information, it is being exposed to increasingly-unusual problem instances, analogously to how the cumulative probability of drawing a value three standard deviations from the mean of a Gaussian increases as



(a) (b)

Figure 4: $\mathbb{R}_{Op}$ and $\mathbb{C}_{Op}$ curves for organizations in Sections 5 and 6 constructed from perfect information.

you draw more samples. For abstractions with high inclusivity, the behaviors from these unusual problem instances inevitably get clustered together with the more common behaviors, essentially creating multimodal distributions for the $\hat{\Delta}_i$s' statistical estimates. Since our ODP computes a $\hat{\Delta}_i$'s incremental impact as a simple expectation of its constituent Δ_is' incremental impacts, multimodal distributions fundamentally violate the assumptions of our ODP's statistical representation, and result in decreased $\mathbb{P}_{Op}$ from the resulting organizations. This argues that an ODP should employ more sophisticated statistical models for representing estimates of high inclusivity influences. An interesting direction to consider for future work would be an automated approach for correcting the ODP's statistical representation in response to influence inclusivity.

Figure 3 also demonstrates that some abstractions are more robust to the effects of information availability than others. For example, the None abstraction is completely immune to these effects and the TCluster abstraction is also exceptionally resilient in this regard. While these abstractions have high inclusivity and thus should be susceptible to information availability effects, they also have extremely low uniformity. Thus, incorporating specialized Δ_is into a $\hat{\Delta}_i$ cannot induce the ODP to permit additional actions, because those actions are already permitted due to the common cases.

6. TASK-DELINEATED ABSTRACTIONS

To summarize the key findings from our analysis in the previous section, the best abstractions are ones that:

Have moderate inclusivity. This provides enough leeway for an ODP to specify nuanced influences where appropriate but is broad enough to permit influences to generalize to the larger problem space.

Have high uniformity. This allows the ODP to more aggressively restrict the agents' local models to a smaller set of behaviors for consideration, which streamlines computational effort thereby improving performance.

Have low variance. Low variance reduces sensitivity in the ODP's search algorithm, resulting in smoother performance curves that better match the Pareto topology.

These observations lead to a high-level strategy of adopting abstractions that segment each problem instance into maximally-sized components that agree on the same behavior with (nearly) the same expected incremental computational and reward impacts. Although such a strategy is not computationally practical as it would involve searching through the space of clusterings, it does suggest a heuristic proxy, which is to group together situations that are collectively pursuing the same outcome. We refer to this heuristic clustering strategy as a **task-delineated abstraction**.

For example, in the fire-fighting domain, this would imply segmenting a problem instance into tasks associated with putting out a specific fire. See Table 2 for the task-delineated abstractions we provided to the ODP for the firefighting domain, where the number of active fires serves as an indicator variable for which task an agent should currently be pursuing. This heuristic leads to abstract influences with moderate inclusivity, high uniformity, and low variance. Inclusivity is moderate because the abstraction allows the ODP to restrict the task-level behaviors of the agents while still allowing information to generalize within the scope of a single task (i.e., provides leeway for agents to use their local expertise to most effectively complete their organizationally designated tasks). Uniformity is high and variance is low if appropriate tasks are identified that cluster similar behaviors with similar incremental impacts.

Figure 5 shows the $\mathbb{P}_{Op}$ curves of these abstractions using the same evaluation methodology as in Section 5, along with the local baseline and the best abstraction from Section 5 (TCluster+Pos). We observe that the heuristically-recommended task-delineated abstraction achieves essentially the same $\mathbb{P}_{Op}$ quality as the TCluster+Pos abstraction, which is unsurprising given that the clustered system times essentially proxy for task-delineation. Notice, however, that because FCount+Pos has lower variance, it is less sensitive to information availability effects (i.e., its three $\mathbb{P}_{Op}$ curves for information quantities are nearly identical), and also exhibits fewer ODP search sensitivities (i.e., the $\mathbb{P}_{Op}$ curves are smoother and monotonically increase over the Pareto topology). As our framework predicts, the FCount abstraction's high variance and low uniformity make it suboptimal.

Finally, it is interesting to recognize that, in retrospect, our previous [13] position-based abstraction is essentially task-delineated for the version of the firefighting domain with only two fires, and as our framework predicts, performed well. That is, since each agent was expected to fight a single fire, there is a single task for each agent.

Abstraction Name	Expected Organizational Inclusivity	Expected Organizational Variance $(10^{-3}\mathbb{R}_{Op}, 10^{3}\mathbb{C}_{Op})$	Description of Θs Constructed from Abstraction
FCount	$\frac{1}{4}$	(3.48, 120)	Number of active fires in a state maps to $\hat{\Delta}_i$.
FCount + Pos	$\frac{1}{200}$	(3.46, 0.11)	Number of active fires and agent's position in a state, together, map to $\hat{\Delta}_i$.

Table 2: Descriptions of task-delineated abstractions (Section 6). Uniformity for each abstraction relies on the specific Pareto characterization, but typically decreases as agent computation becomes less costly. For completeness, we included the FCount abstraction despite our framework predicting that it has poor performance characteristics. We also created a clustered position variant, but it was qualitatively identical to FCount+Pos.

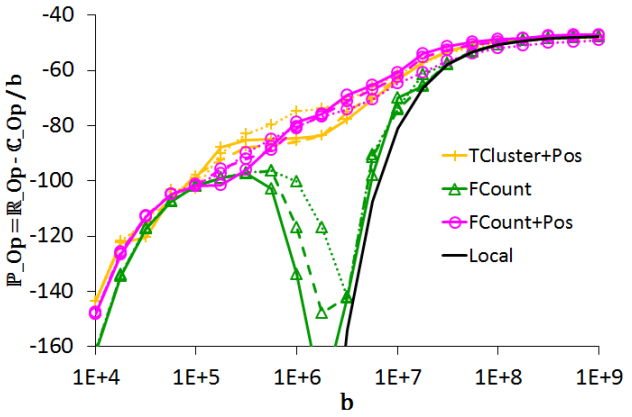


Figure 5: $\mathbb{P}_{Op}$ curves for task-delineated abstractions alongside bounding abstractions from Section 5. Solid, dashed, and dotted curves correspond to perfect information, 2/3 information, and 1/3 information respectively.

7. CONCLUSION

In this paper, we systematically examined how abstraction choices can impact an ODP and the subsequent operational performance of the design it produces. We constructed a framework for characterizing abstract organizational influences along key dimensions, and empirically evaluated several families of abstract organizational influences. In our subsequent analysis, we identified the heuristic that the most effective abstractions tend to have moderate inclusivity, high uniformity, and low variance, and using this as a foundation, constructed a task-delineated abstraction that achieves the desired characteristics, and in turn compares favorably to the other abstraction choices in our evaluation domain.

This work points to several other research directions in abstraction for organizational design. Beyond extensions to many-to-many abstractions mentioned in Section 4.3, future work could also investigate the possibility of multiple abstractions which could allow the ODP to specify detailed, low-inclusivity influences when it is important to differentiate behaviors, and then specify better generalizing, high-inclusivity influences elsewhere. Other directions include investigating the relationship between organizational abstraction and flexibility, and identifying more robust statistical representations for modeling the incremental impact of high-inclusivity influences.

8. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their feedback. This work was supported in part by NSF grant IIS-0964512.

9. REFERENCES

[1] C. Amato, G. D. Konidaris, and L. P. Kaelbling. Planning with macro-actions in decentralized POMDPs. In *AAMAS*, pages 1273–1280, 2014.

[2] R. Becker, S. Zilberstein, V. Lesser, and C. Goldman. Solving transition independent decentralized Markov decision processes. *JAIR*, 22:423–455, 2004.

[3] D. S. Bernstein, E. A. Hansen, and S. Zilberstein. Bounded policy iteration for decentralized POMDPs. In *IJCAI*, pages 1287–1292, 2005.

[4] V. Dignum, J. Vázquez-Salceda, and F. Dignum. Omni: Introducing social structure, norms and ontologies into agent organizations. In *Programming Multi-Agent Systems*, pages 181–198, 2005.

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

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

[8] P. Poupart and C. Boutilier. Bounded finite state controllers. In *NIPS*, pages 823–830, 2003.

[9] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

[10] S. Sardina, L. de Silva, and L. Padgham. Hierarchical planning in BDI agent programming languages: A formal approach. In *AAMAS*, pages 1001–1008, 2006.

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

[12] J. Sleight and E. H. Durfee. Organizational design principles and techniques for decision-theoretic agents. In *AAMAS*, pages 463–470, 2013.

[13] J. Sleight and E. H. Durfee. Multiagent metareasoning through organizational design. In *AAAI*, 2014.

[14] 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.

[15] 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.

[16] S. J. Witwicki. *Abstracting Influences for Efficient Multiagent Coordination Under Uncertainty*. PhD thesis, University of Michigan, 2010.