Minimal Mental Models

David V. Pynadath and Stacy C. Marsella
Information Sciences Institute, University of Southern California
4676 Admiralty Way, Marina del Rey CA 90292 USA
{pynadath,marsella}@isi.edu

Abstract

Agents must form and update mental models about each other in a wide range of domains: team coordination, plan recognition, social simulation, user modeling, games of incomplete information, etc. Existing research typically treats the problem of forming beliefs about other agents as an isolated subproblem, where the modeling agent starts from an initial set of possible models for another agent and then maintains a belief about which of those models applies. This initial set of models is typically a full specification of possible agent types. Although such a rich space gives the modeling agent high accuracy in its beliefs, it will also incur high cost in maintaining those beliefs. In this paper, we demonstrate that by taking this modeling problem out of its isolation and placing it back within the overall decision-making context, the modeling agent can drastically reduce this rich model space without sacrificing any performance. Our approach comprises three methods. The first method clusters models that lead to the same behaviors in the modeling agent's decision-making context. The second method clusters models that may produce different behaviors, but produce equally preferred outcomes with respect to the utility of the modeling agent. The third technique sacrifices a fixed amount of accuracy by clustering models that lead to performance losses that are below a certain threshold. We illustrate our framework using a social simulation domain and demonstrate its value by showing the minimal mental model spaces that it generates.

Introduction

Agents face the challenge of forming and updating their mental models of each other in a wide range of multiagent domains. Research in plan recognition has produced an array of techniques for modeling a planning agent and forming a belief about what its goals and intentions are, so as to predict its future actions (Schmidt, Sridharan, & Goodson 1978; Kautz & Allen 1986). User modeling faces a similar problem in trying to understand and anticipate the needs of human users interacting with a software system (Jameson 1995). Agents working together as teams must maintain beliefs about their teammates' status (Kaminka, Pynadath, & Tambe 2002). Social simulation may require agents with a theory of mind about the other agents in their society (Pynadath & Marsella 2005). In games of incomplete information, each player faces uncertainty about the payoffs that the other players will receive (Fudenberg & Tirole 1991).

In these areas, forming mental models is treated as a separate subproblem within the overarching decision-making context of the agent. The modeling agent starts from an initial set of possible models for the other agents, whether in the form of plan libraries in plan recognition, possible mental models in social simulation, private types in games of incomplete information, etc. As the modeling agent interacts with the other agents, it updates that belief based on its observations of their behavior. The modeling agent then uses its mental models of the other agents to make informed decisions based on expectations of what they will do.

For example, in a social simulation of a classroom, a teacher deciding how to maintain discipline would find it useful to keep track of which students (dis)like each other. If she is also planning a picnic lunch, then she might also want to keep track of which students prefer pizza to hamburgers. In general, enriching the mental models that the teacher has of her students enables her to make better decisions. On the other hand, it is harder for her to maintain correct beliefs over the richer models. Intuitively, we expect a diminishing return on enriching the mental models, where adding more details offers less gain in accuracy, while increasing computational cost. For example, while the teacher could also keep track of her students' musical preferences, she would expect little benefit from doing so.

Agent modeling approaches vary in their method for choosing the appropriate refinement of the mental model space. Typical approaches rely on a human designer to implicitly define the level of detail (e.g., deciding which features, goals, plans, etc. to represent). These approaches focus on the representational needs regarding the modeled agent, without any consideration of the decision-making needs of the modeling agent. Thus, they run the risk of generating either an overly detailed space of mental models that will unnecessarily burden the modeling agent, or an overly coarse space that will lead to suboptimal performance.

In this paper, we demonstrate that we can choose a minimal, but sufficient, space of mental models by taking the problem of modeling others out of its isolation and placing it within the overall decision-making context of the modeling agent. The agent can then automatically derive a space of mental models according to an informed analysis of the cost-benefit tradeoffs. In most domains, agents can expect that this analysis will allow them to drastically reduce the original full mental model space, without overly sacrificing performance. Furthermore, the generated mental model space will be optimized for the individual modeling agent's needs, rather than having multiple modeling agents using the same space of models for the same observed agent.

We present three methods that we have implemented to quantify the value of including a candidate mental model: *Behavior Equivalence*, where the modeling agent clusters models that lead to the same behavior of the modeled agent; *Utility Equivalence*, where the modeling agent clusters models that may lead to different behaviors, but produce equally preferred outcomes in terms of utility; and *Approximate Utility Equivalence*, where the modeling agent clusters models that lead to performance losses below a certain threshold, sacrificing a fixed amount of accuracy.

Modeling Other Agents

Across the various multiagent research areas already mentioned (and even within each area itself) researchers have applied a wide variety of possible representations. We present a methodology using an abstract agent modeling framework that is general enough to cover these approaches, as well as other decision-making procedures in the literature. When applying our methodology to a specific domain, these components would become specialized to the particular framework used for the agents in that domain.

Agent Notation

At the most general level, an agent model is a triple, $\langle B, A, U\rangle$. B is the set of possible beliefs that the agent can have about its environment (including the other agents), A is the set of available actions, and U is a utility function over outcomes. We use the same structure to represent both the actual agents and the mental models they have of each other. Thus, we represent the multiagent system itself as a set of real agents, $\{m_i\}_{i=1}^N$. For each such agent model, $m_i = \langle B_i, A_i, U_i\rangle$, the beliefs, B_i, consist of a set of possible belief states including possible beliefs over mental models, M_{ij}, that represent what agent i can think of agent j.

The modeling agent wishes to minimize this space, M_{ij}. In particular, we want an algorithm that computes the expected utility derived by the modeling agent i when using the set of mental model spaces, $\{M_{ij}\}_{j=1}^N$, for all of the agents in the system (potentially including itself). We define the behavior of an agent as a policy, $\pi : B \rightarrow A$, out of a set of possible policies, Π. Any agent architecture will include an algorithm for translating an agent into such a policy, π. We will abstract this procedure into a generic function $\textsc{Solve}$: $M \rightarrow \Pi$, that takes an agent model (whether real or subjective) and returns that model's policy of behavior.

Example Domain

We have taken our example domain from a scenario in childhood aggression, modeled within PsychSim, a multiagent social simulation tool (Pynadath & Marsella 2005). There are agents for three students: a bully, his victim (i.e., the student he focuses his aggression on), and an onlooking student to whom the bully looks for affirmation. There is also a teacher who wishes to prevent any incidents of aggression. The teacher can deter the bully by doling out punishment. We focus on the problem facing the bully agent, whose decision on whether to pick on his victim must consider the possible punishment policy of the teacher. This domain is complicated enough to demonstrate a rich interdependency between the bully's mental models of others and its own decision making. However, it is also simple enough to highlight only the most relevant issues in constructing such mental models and to support a broad suite of experiments.

Utility The bully agent uses a decision-theoretic model of preferences, so it decides whether or not to pick on his victim through a finite-horizon evaluation of expected utility. There are three components of the bully's utility function: (1) a desire to increase his power, which decreases when he is punished; (2) a desire for affirmation from the onlooking student, which increases when the onlooker laughs along; and (3) a desire to decrease the victim's power, which decreases when the bully picks on him (as well as when the onlooker laughs at him). We define the utility function as a linear combination of these three components, so that we can represent the bully's type as a triple of coefficients from $[0, 1]$. To simulate the behavior of a bully whose aggression is intended to gain the approval of his peers, we would use an agent with a higher weight for the second of the utility components (increase affirmation). On the other hand, to simulate a more sadistic bully, we would use a higher weight for the third (decrease victim's power).

The teacher also has three components to her utility function, corresponding to her desire to increase the power of the three students. The teacher thus has a disincentive for punishing anyone unless doing so will deter acts that would reduce the victim's power even more. A perfectly fair teacher would give equal weight to the three students' power. A bully feeling persecuted by the teacher may think that she favors the victim's power over his own. On the other hand, a bully may feel that the teacher shares his dislike of the victim (i.e., has a lower weight for increasing the victim's power).

We focus on the problem of the bully's modeling of the teacher, so we fix the onlooker to value his own power (i.e., he does not want to be punished), while also wanting to decrease the power of the victim out of dislike (i.e., he enjoys laughing at the victim when the bully picks on him). Furthermore, both the bully and teacher know the correct model of the onlooking student (i.e., $|M_O| = |M_{BO}| = |M_{TO}| = 1$).

Actions The teacher has 7 options in her action set, A_T. She can do nothing; she can scold the bully, onlooker, or the entire class; or she can punish the bully, onlooker, or the entire class. Punishing a student causes a more severe decrease in a student's power than simply scolding. The onlooking student has 2 options in his action set, A_O: laugh at the victim, or do nothing. The bully has 2 actions in his action set, A_B: pick on the victim or do nothing.

Policies To reduce the domain to its most essential aspects, the bully's policy, $\pi_B : M_{BO} \times M_{BT} \rightarrow A_B$, is a function of his mental model of the onlooker and teacher. Given that the onlooker has only one possible mental model, the policy space for the bully, Π_B, contains $|A_B|^{|M_{BT}|}$ distinct policies. Thus, the complexity of the bully's problem of choosing his correct policy is highly dependent on the number of mental models that he must consider for the teacher.

Similarly, the onlooker's policy, $\pi_O : M_{OB} \times M_{OT} \rightarrow A_O$, depends on only his mental model of the bully and the teacher. The bully assumes that the onlooker knows the bully's true model and that the onlooker's mental model of the teacher is the same as the bully's. Thus, in this current investigation, we focus on only one entry in π_O, namely the one where $m_{OB} = m_B$ and $m_{OT} = m_{BT}$, where there are only two possible values: laughing at the victim or not.

We must also specify what the bully expects the teacher to do, which depends on not only her mental models of the students, but also on the prior actions of the students. In other words, the teacher may perform a different action when the bully picks on the victim than when he does not. Thus, the policy, $\pi_T : M_{TB} \times M_{TO} \times A_B \times A_O \rightarrow A_T$, is much more complicated than that of the students. However, we again simplify this model by having the bully assume that the teacher knows the correct model of him (i.e., $m_{TB} = m_B$) and shares his mental model of the onlooker (i.e., $m_{TO} = m_{BO}$). With this restriction, the teacher's relevant policy space, Π_T, contains $|A_T|^{|A_B|\cdot|A_O|} = 2401$ distinct punishment policies for the teacher to consider. Thus, even with our simplifying assumptions, there still remains a large space of possible behaviors for the teacher.

Solution Mechanism We use boundedly rational agents to simulate the behavior of all of the entities in our social scenario. Thus, the bully's SOLVE algorithm performs a forward projection over his possible actions, computes the expected utility, and chooses the action with the highest value. The forward projection computes the total utility over the bully's action, the onlooker's subsequent response, and the resulting punishment action (or lack thereof) taken by the teacher. To determine the teacher's policy, the bully applies a SOLVE method from the teacher's perspective that exhaustively tries all possible policies in Π_T, computes the best response policies for the bully and onlooker, and then chooses the best policy based on her expected utility over the entire interaction. Given this policy for the teacher, the bully and onlooker can then choose their policies as best responses.

With this solution procedure, we can completely specify the bully's mental model of the teacher in terms of the three utility weights that the bully attributes to her. In other words, our initial space of candidate mental models, M_{BT}, contains one model for every vector of weights, $\vec{w} = [w_B, w_O, w_V]$, subject to the constraint that $\sum_{w \in \vec{w}} w = 1$. This continuous space generates an infinite number of mental models for the bully to potentially consider. In practice, the bully can discretize the space by considering only points along a grid. For a granularity of 0.1, there are 66 possible vectors: $[0.0, 0.0, 1.0]$, $[0.0, 0.1, 0.9]$, $[0.0, 0.2, 0.8]$, $\dots$, $[0.9, 0.1, 0.0]$, $[1.0, 0.0, 0.0]$. For a granularity of 0.01, there are 5151 possible vectors, while for a granularity of 0.5, there are only 6.

In choosing the discretization of this mental model space, the bully must consider the interdependency between his own decisions and the decisions he expects the teacher to make. For example, if he picks on the victim, he is more likely to be severely punished by a teacher for whom the victim is a pet (i.e., for which w_V is high), but he would be more likely to escape punishment if he himself is a favorite of the teacher (i.e., if w_B is high). Thus, there is clearly some value to be gained by maintaining differential mental models of the teacher. However, it is unlikely that real-life bullies juggle 66 (let alone 5151) possible mental models of their teachers in their heads. In fact, it is unclear whether even a space of 6 models is more than the bully needs to consider. Furthermore, it is likely that the bully needs more fine-grained distinctions in certain regions of the model space, so a uniform discretization (regardless of the granularity) will be suboptimal. Unfortunately, it is unclear what granularity of discretization produces the minimal set of mental models that is still sufficient for the bully's decision-making needs.

This scenario is illustrative, and there are clearly many dimensions along which we could enrich it. For example, we could introduce state dependencies (e.g., the weaker the victim, the more damage done by picking on him). However, while these additional wrinkles would change the particular answers provided by our methodology, they would not change the *ability* of the methods presented in the following sections to provide such answers. In fact, additional complications would only introduce *more* dimensions to discretize, making the need for an automated method for minimizing mental model spaces all the more urgent. Our core methodology presents a very general method for quantifying the value of candidate mental models even in the face of these additional complications. Therefore, we have removed as many extraneous domain features as possible, so as to be able to provide the clearest illustration of the methods and how they can be applied to any multiagent domain.

Behavior Equivalence

The modeling agent's goal is to find a minimal set of mental models that it needs to consider for the other agents. In looking for possible bases for such minimization, we observe that the modeling agent's decisions often depend on only the *behavior* of the agents being modeled. Agents model the hidden variables of others so as to generate expectations of their resulting behavior, but given the behavior of others, an agent's decision making is often conditionally independent of the internal reasoning behind it. For example, in agent teamwork, the mental states of the individual members have no direct effect on performance; only the decisions (actions, messages, etc.) derived from those mental states matter. Similarly, in games, the payoffs received by the agents depend on only the moves chosen by the players. In social simulations, the agents cannot read each others' minds, so they can base their decisions on only their observable behaviors. Therefore, regardless of what underlying parameters govern the modeled agent's decision-making, its eventual behavior is what has an impact on the modeling agent.

Behavior Equivalence Algorithm

This observation forms the basis for our first method for constructing a minimal space of mental models. If two mental models produce the same behavior for the modeled agent, then making a distinction between them does not help the modeling agent. Therefore, it needs to include only one of them for consideration. It can do so by computing the policies corresponding to any candidate mental models and inserting only those that generate distinct policies. Algorithm 1 allows the agent to incrementally construct its mental model space, M, by considering each candidate model, m, in turn and inserting only those for which BehaviorEquivalence returns `False`.

Algorithm 1 BehaviorEquivalence(M, m)

```
1: for all $m' \in M$ do
2:     if Solve$(m)$ = Solve$(m')$ then
3:         return True
4: return False
```

For many domains, the repeated invocations of the Solve function can be computationally intensive, but there is plenty of opportunity for specialization of Algorithm 1. For example, if the mental models correspond to points in a metric space (as in our social simulation domain), it should be possible to compare mental models to only their immediate neighbors in that space. Furthermore, even if specializing the algorithm is insufficient, there are many opportunities for approximation as well. For example, one could easily re-write the exhaustive loop in Line 1 to instead randomly sample only a subset of models to compare against the new candidate for behavior equivalence. Similarly, one could approximate the equivalence test in Line 2 to test the policies for equality over only a randomly selected subset of entries.

Behavior Equivalence Results

The bully agent can generate a set of mental models that is minimal with regard to behavior equivalence, but the policy chosen by the teacher also depends on her model of the bully. For example, different bullies may be more afraid of a teacher punishing the whole class because of him than of being punished by himself. We thus performed a behavior equivalence construction of the mental model space for different types of bullies. To do so, we discretized the space of possible (real) bullies in the same way that we discretized the space of possible mental models of the teacher. Thus, we represent different types of bullies by different vectors of utility weights, $\vec{w} = [w_B, w_O, w_V]$, and discretize the set of possible types into 66 distinct such vectors, $[0.0, 0.0, 1.0]$, $[0.0, 0.1, 0.9]$, $[0.0, 0.2, 0.8]$, $\ldots$, $[0.9, 0.1, 0.0]$, $[1.0, 0.0, 0.0]$.

In our experiments, we had each of the 66 possible bully types consider the 66 candidate mental models for the teacher, also generated with a granularity of 0.1. We gave the teacher and onlooker the correct model of the bully and of each other when running Algorithm 1. 8 types of bullies constructed a mental model space that was minimal with respect to behavior equivalence (denoted by M^b_{BT}) that had only 4 of the original 66 candidate mental models. The other 58 types of bullies constructed a space of 5 models.

Behavior equivalence provides a clear benefit to these types of bully agents. In particular, it is notable that, although the 66 types of teachers had 2401 policies to choose from, a specific bully could expect to come across only 4 or 5 distinguishable teacher behaviors. In fact, looking across the results for all of the possible bully types, there were only 8 policies that were *ever* selected by the teacher in the $66 \cdot 66 = 4356$ bully-teacher combinations. The reason that so much of the teacher's policy space is undesirable for her is that the bully's behavior is constrained by his utility. For example, regardless of where in our utility space he is, the bully always prefers not being punished to being punished. Therefore, it would never make sense for the teacher to adopt a policy of punishing the bully if he does nothing to the victim and doing nothing to him if he does.

Utility Equivalence

There are some multiagent domains where the modeling agent derives some direct utility from the values of hidden variables. For example, some teams may receive additional utility when all of the team members have the same beliefs about the current team plans, even if such coordination of beliefs is not required for achieving the desired behavior. Alternatively, in our social simulation, the teacher may prefer being liked by her students, rather than feared, even if both cases produce complete obedience. In such cases, behavior equivalence's focus on only observable actions may ignore some necessary distinctions in the mental model space. However, it is still safe to assume that the modeled agent matters only in so far as it affects the modeling agent's expected utility. The modeling agent is thus completely indifferent between different mental models that produce the same expected utility in its own execution.

Utility Equivalence Algorithm

This observation leads to our second method for generating a minimal mental model space. If the modeling agent does not lose any expected utility when using a particular mental model when the correct model is actually another, then making a distinction between the two models does not help. Therefore, it is safe for the modeling agent to remove one of them from consideration. It can do so by computing its expected utility based on the policies corresponding to each of the possible mental models (of the modeled agent) and insert only those that lead to lost utility when mistaken for another already under consideration. Like Algorithm 1 for behavior equivalence, Algorithm 2 allows the agent to construct its mental model space, M, incrementally, by considering each candidate model, m, in turn and inserting only those for which UtilityEquivalence returns `False`.

While behavioral equivalence requires only the modeled agent's policy, utility equivalence requires the further computation of the modeling agent's own best response to that policy (where its own model is m_{modeler}). Line 4 shows that the modeling agent computes the expected utility (u_{wrong}) it will derive if it solves for its policy assuming

Algorithm 2 UTILITYEQUIVALENCE(M, m)

```
1: for all $m' \in M$ do
2:     $\pi \leftarrow \text{SOLVE}(m)$, $\pi' \leftarrow \text{SOLVE}(m')$
3:     $u_{\text{right}} \leftarrow EU\left[\text{SOLVE}\left(m_{\text{modeler}} \middle| m\right) \middle| \pi\right]$
4:     $u_{\text{wrong}} \leftarrow EU\left[\text{SOLVE}\left(m_{\text{modeler}} \middle| m'\right) \middle| \pi\right]$
5:     if $u_{\text{right}} - u_{\text{wrong}} \leq 0$ then
6:         return True
7: return False
```

that the modeled agent is of type m', when it is actually of type m. Line 3 computes its expected utility (u_{right}) when it holds the correct mental model, m. If the first is lower than the second, then the agent must include m in its minimal mental model space.

The inequality in Line 5 accounts for the possibility that an incorrect mental model may produce an actual utility *gain* when the agent being modeled, in turn, has an incorrect model of the modeling agent. Over time, if the agent being modeled updates its belief about the modeling agent, then such a utility gain is unlikely, because the modeled agent could eventually settle on a best response to the modeling agent's misconception. However, in the transient behavior, the modeled and modeling agents may inadvertently act in ways that improve the modeling agent's utility, despite the error in mental models.

Algorithm 2 adds another round of calls to the SOLVE function beyond what behavioral equivalence required. The additional cost comes with the benefit of a guaranteed minimality, in that removing any mental model from the generated space will cause the modeling agent to suffer a loss in expected utility.

Utility Equivalence Results

To generate the bully's mental models of the teacher that were minimal with respect to utility equivalence, we followed the same experimental setup as for behavior equivalence. The 66 types of bully agents ran Algorithm 2 over the 66 candidate mental models. For this specific scenario, behavior equivalence implies utility equivalence, as the bully agent derives no direct utility from the teacher's intrinsic parameters. We can thus cluster the utility equivalence results according to the further reductions in mental model space achieved from M_{BT}^{b}. Of the 58 bully types with $\left|M_{BT}^{b}\right| = 5$, 11 types of bullies constructed a minimal mental model space with respect to utility equivalence that had only 2 of the 66 candidate teacher models, while the other 47 types constructed a minimal mental model space of size 4. Of the remaining 8 bully types with $\left|M_{BT}^{b}\right| = 4$, all of them generated a space of 3 mental models. Furthermore, for every type of bully, the mental model spaces constructed by utility equivalence (denoted M_{BT}^{u}) are all strict subsets of those constructed by behavior equivalence.

Some cases of utility equivalence occur for bullies with extreme utility weights. For example, to a bully who cares about only hurting the victim (i.e, $\vec{w} = [0.0, 0.0, 1.0]$), mental models that differ on whether he himself gets punished are equivalent, because he does not care about the decrease in his own power. However, mental models that differ on whether or not the *onlooker* gets punished are not equivalent, because he wants the onlooker to laugh at the victim as well, to maximize the damage inflicted on the victim.

Utility equivalence sometimes occurs in this experiment when an incorrect mental model of the teacher increases the bully's expected utility. For example, two mental models of the teacher may differ regarding whether they would punish the onlooker. From the bully's point of view, if the onlooker laughs regardless of the teacher's policy, then the bully does not care whether the onlooker is punished. Thus, while these two mental models produce different teacher behaviors, mistaking one for the other does not decrease the expected utility of the bully, who is then justified in ignoring the distinction between them.

Approximate Utility Equivalence

A mental model space constructed according to utility equivalence is truly minimal with respect to the modeling agent's decision making. Any further clustering of mental models will cost the modeling agent utility. However, the modeling agent can reduce its cost of maintaining beliefs and finding a policy over the mental model space by also clustering those models that sacrifice a small amount of utility.

Approximate Utility Equivalence Algorithm

This observation leads to our third method for reducing the space of possible mental models. We can easily adapt our utility equivalence algorithm to be tolerant of any utility loss below some positive threshold.

Algorithm 3 UTILITYAPPROX(M, m, θ)

```
1: for all $m' \in M$ do
2:     $\pi \leftarrow \text{SOLVE}(m)$, $\pi' \leftarrow \text{SOLVE}(m')$
3:     $u_{\text{right}} \leftarrow EU\left[\text{SOLVE}\left(m_{\text{modeler}} \middle| m\right) \middle| \pi\right]$
4:     $u_{\text{wrong}} \leftarrow EU\left[\text{SOLVE}\left(m_{\text{modeler}} \middle| m'\right) \middle| \pi\right]$
5:     if $u_{\text{right}} - u_{\text{wrong}} \leq \theta$ then
6:         return True
7: return False
```

The pseudocode in Algorithm 3 is written to support execution with a fixed threshold in mind. Alternatively, one could perform Lines 1–4 and *then* choose an appropriate threshold, θ, to reduce the space to an appropriate size. In other words, one would first profile the possible errors that would be derived from incorrect mental models before choosing a clustering. One could also easily vary the computation to use error measures other than expected utility. For example, one might be interested in worst-case utility loss instead of expected-case. Simply replacing the expectation in Lines 3 and 4 with a maximization would make the desired adjustment. There are any number of variations that would similarly modify the optimality criterion used in weighing the utility lost from the mistaken mental model.

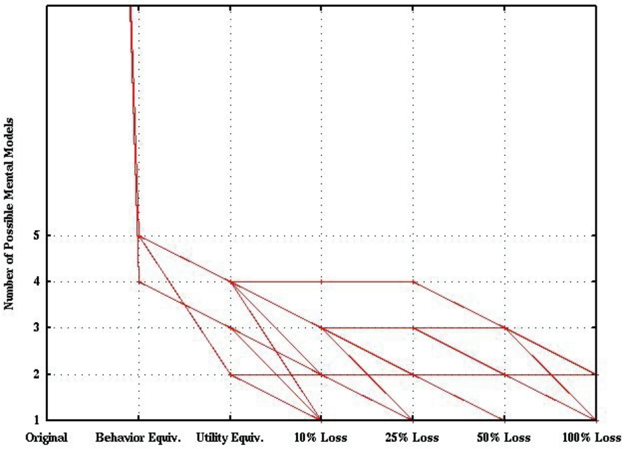

Figure 1: Size of model spaces vs. increasing leniency for utility loss, across all types of bully agents.

Approximate Utility Equivalence Results

To construct the bully's minimal mental models for the teacher according to approximate utility equivalence, we followed the same experimental setup as for the other two methods. Figure 1 shows the results across our three methods for mental model space construction. Each path from left to right represents the size of the mental model space for at least one possible type of bully as we raise its tolerance for utility loss. At the y-axis, all of the bully agents have all 66 of the candidate mental models. Then we see that these agents need to consider only 4 or 5 models, using only the behavior equivalence method. The next point shows that the bully agents have spaces of 2–4 mental models when using only the utility equivalence method. Continuing along a path to the right represents the further reduction in the mental model space that comes with clustering mental models that cost less that the given threshold of expected utility.

Figure 2 shows the number of bully types (i.e., utility weight combinations) that follow each of the possible paths. For example, there are 7 (the "#" column) bully types that follow a path that leads to a mental model space of size one with only 10% loss of expected utility. If bully agents of this type are willing to tolerate a small utility loss, they can do away with modeling the teacher altogether!

At the opposite end of the spectrum, there is one bully type that follows the upper envelope of the graph. For this bully type, Figure 1 shows that utility equivalence (UE) constructs a mental model space of size 4, down from the size 5 of the space using only behavior equivalence (BE). However, we see that even if the bully is willing to tolerate a loss of 25% of its expected utility, it still needs this full space of 4 models. If it wants to reduce its mental model space by even one element, it can incur an up to 50% loss in expected utility if it is wrong. This bully type is also one of 14 in our sample space for which tolerating even 100% utility loss is not sufficient to warrant eliminating mental modeling together, as using the wrong mental model will lead to *negative* utility.

BE	UE	10%	25%	50%	100%	#	BE	UE	10%	25%	50%	100%	#
4	3	1	1	1	1	1	5	4	2	2	1	1	9
4	3	2	1	1	1	1	5	4	2	2	2	1	10
4	3	2	2	1	1	3	5	4	2	2	2	2	3
4	3	2	2	2	1	3	5	4	3	1	1	1	2
5	2	1	1	1	1	1	5	4	3	2	2	1	4
5	2	2	2	1	1	1	5	4	3	2	2	2	2
5	2	2	2	2	1	2	5	4	3	3	2	1	3
5	2	2	2	2	2	7	5	4	3	3	2	2	1
5	4	1	1	1	1	5	5	4	3	3	3	1	1
5	4	2	1	1	1	6	5	4	4	4	3	2	1

Figure 2: Number of bully types for each progression of mental model sizes with increasing leniency for utility loss.

Discussion

While the exact numbers in Table 2 are specific to our example domain, they provide a concrete demonstration of our general ability to quantify the value of mental models to the modeling agent. To make the final decision, the agent must consider the computed value of the mental model space along with the cost of performing the actual model update and decision making during execution. This tradeoff will, in part, depend on the agent framework, but the payoff can be considerable. For example, as already described, the space and time complexity of the policy solution and model update procedures of the modeling agent can grow dramatically with the number of mental models to consider. For example, our bully agent employs a decision-theoretic approach by computing a probability distribution over the possible mental models and by performing policy iteration to determine its optimal behavior. The computational complexity of both algorithms grows exponentially with the number of possible mental models. Such an optimizing approach is possible when using the minimal mental model spaces of size 2–4, but uninformed discretizations (e.g., 66 models at granularity 0.1) would require approximate methods to be able to maintain the same practical computation times.

We should stress that the general problem of determining minimal mental model spaces, as well as the approaches we present to solve that problem, are not limited to our chosen mental model representation. We can examine the general problem of determining minimal mental model spaces in the context of other agent representations as well. At a very general level, Interactive Partially Observable Markov Decision Problems (Interactive POMDPs) represent mental models as *types* that capture the beliefs and planning processes of other agents (Gmytrasiewicz & Doshi 2005). Given a set of possible types for the teacher, a bully agent can solve an Interactive POMDP to derive an optimal policy for his behavior, based on a continually updated probabilistic belief over which type she actually was. In this context, our work solves the unaddressed problem of determining the minimal set of possible types for the bully to consider.

Most other existing work in agent modeling has no explicit enumeration of the possible mental models to consider. Such approaches typically start from a domain representation that only implies the possible states of the agent being modeled. Much of the early work in plan recognition used first-order logic to represent the plan libraries of the modeled agent and standard inference mechanisms to derive a belief for the modeling agent (Kautz & Allen 1986). Later work used a probabilistic model of the plan library and applied Bayesian networks to infer a distribution over possible mental models (Charniak & Goldman 1993; Goldman, Geib, & Miller 1999). These plan libraries implicitly specify a set of mental models in the form of the possible combinations of plans that the modeled agent may be actively pursuing at any one time.

Using such approaches in the classroom scenario, the space of mental models of the teacher is the set of possible active plan configurations implied by the plan library. A bully agent faces the problem of determining when to stop enriching its beliefs about the teacher's plan library (e.g., by including the subplans of an existing plan). The plan library is typically a generative model of the planning behavior of the agent being modeled. However, different students will have different priorities in modeling the teacher's plans. For example, while the bully would be especially concerned about the teacher's punishment plans, a better behaved student might be more concerned about her educational plans. Given that the number of possible mental models is roughly exponential in the possible plans, it does not make sense for all of the students to use the same, complete plan library of the teacher. Whether the plan library is encoded by human designers or learned by the modeling agent itself, our algorithms allow different agents to automatically make different decisions about their minimal set of mental models.

Other existing agent modeling techniques forego explicit modeling of the plan library as well (Hong 2001). These methods instead start with a set of domain axioms that represent the possible states and actions of the modeled agent and its environment. Again, the number of mental models grows roughly exponentially with the number of state predicates and actions that are included in the domain model. A bully agent using such a representation can use our algorithms to decide what atoms and axioms to include for a minimal domain model that is still sufficient for capturing the relevant details of the teacher's planning state.

Our methodology can also potentially create more psychologically plausible social simulations. In our experiments, the bully agents who were more attention-seeking (i.e., higher w_O) derived less value from the more complete mental model spaces for the teacher. Our characterization of bully types is consistent with the psychological findings that one can characterize different types of childhood aggression by the different goals that bullies have (Schwartz 2000). Thus, we can use our algorithms to explore the mental model spaces that we derive from those different goals and validate them against experimental data. Having validated the agents against such data, we can generate more confidence in the realism of the simulation.

Conclusion

At a higher level, the result of this investigation provides a key insight into the impact of social interaction on the design of multiagent systems. As designers, our immediate reaction is to view such interactions as complicating the problem of deriving appropriate multiagent behavior. However, as our results show, the interplay between the decision-making and modeling efforts of the individual agents is also highly *constraining* on that behavior. For example, out of the 2401 possible policies for the teacher, only 8 were ever desirable when interacting with our 66 types of bullies. When we view the problem of modeling other agents through the subjective lens of the modeling agent's own decision-making, we gain a utility metric that we can use both to restrict the scope of the modeling problem and to derive algorithms to solve it.

We used the utility metric to design algorithms that quantify the value of distinctions made within the mental model space and reduce that space accordingly. An agent can also use this same metric to derive a mental model space from scratch, simply by quantifying the value of *adding* mental models to the space of consideration. In this manner, our metric allows an agent designer to isolate those aspects of the mental models that are most relevant to the agent. We expect the algorithms to give such designers novel insight into the nature of their domains and to minimize the computational complexity of modeling other agents in all multiagent domains where such modeling is beneficial.

References

Charniak, E., and Goldman, R. P. 1993. A Bayesian model of plan recognition. *Artificial Intelligence* 64(1):53–79.

Fudenberg, D., and Tirole, J. 1991. *Game Theory*. MIT Press.

Gmytrasiewicz, P., and Doshi, P. 2005. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research* 24:49–79.

Goldman, R. P.; Geib, C. W.; and Miller, C. A. 1999. A new model of plan recognition. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 245–254.

Hong, J. 2001. Goal recognition through goal graph analysis. *Journal of Artificial Intelligence Research* 15:1–30.

Jameson, A. 1995. Numerical uncertainty management in user and student modeling: An overview of systems and issues. *User Modeling and User-Adapted Interaction* 5(3-4):193–251.

Kaminka, G.; Pynadath, D. V.; and Tambe, M. 2002. Monitoring teams by overhearing: A multi-agent plan-recognition approach. *Journal of Artificial Intelligence Research* 17:83–135.

Kautz, H. A., and Allen, J. F. 1986. Generalized plan recognition. In *Proceedings of the National Conference on Artificial Intelligence*, 32–37.

Pynadath, D. V., and Marsella, S. C. 2005. PsychSim: Modeling theory of mind with decision-theoretic agents. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1181–1186.

Schmidt, C.; Sridharan, N.; and Goodson, J. 1978. The plan recognition problem: An intersection of psychology and artificial intelligence. *Artificial Intelligence* 11:45–83.

Schwartz, D. 2000. Subtypes of victims and aggressors in children's peer groups. *Journal of Abnormal Child Psychology* 28:181–192.