

situation is presented, the model calculates the similarity between the current situation and each episode in long-term memory, which determines the proportion that each remembered episode contributes to the recognition process. A course of action is selected and the outcome (successful or unsuccessful) is determined. The episode, including situation cues, course of action, and outcome, then goes into long-term memory.

Generally, courses of action selected by the RPD model determine the flow of control in the Micro Saint Sharp environment model at important branching points. They can also be used to set parameters or variables in the environment model. To create a decision model, the modeler needs to 1) identify courses of action, 2) map simulation (task network) variables to cues in the RPD model, and 3) define successful outcomes for each course of action.

3. The Models

In this next section, we will discuss three types of models that we have built with the MS/RPD approach: categorization models, dynamic models, and a larger-scale exploratory model of a complex militarily-relevant task.

3.1 Categorization Models

Several of the models we have built can be classified as categorization models. The first we will discuss here is a model of the Brunswik Faces task (see Warwick & Fleetwood, 2006). The Brunswik faces are a set of stimuli that follow the 5-4 category structure, often used to study categorization (Smith & Minda, 2000, Gluck et al., 2001). Each face stimulus is defined by four binary features (see Figure 1): eye height (high or low), eye separation (wide or narrow), nose length (long or short), and mouth height (high or low). Combining these four features yields 16 possible faces.

These 16 faces can be grouped according to their features into two categories, A and B, defined by a complex rule that is not easily verbalized. The 5-4 category structure is intended to make categorization difficult, as only combinations of features are diagnostic. Participants are trained on nine of the faces, and once they meet a performance criterion, they are tested on the full set of 16 faces, including the seven faces that they have not seen before, to test their ability to generalize the categorization rule.

We presented the nine training faces (each a set of four cue values), to the model in randomized order within training block. In one condition, the model was trained on a set number of training blocks. In the other condition,

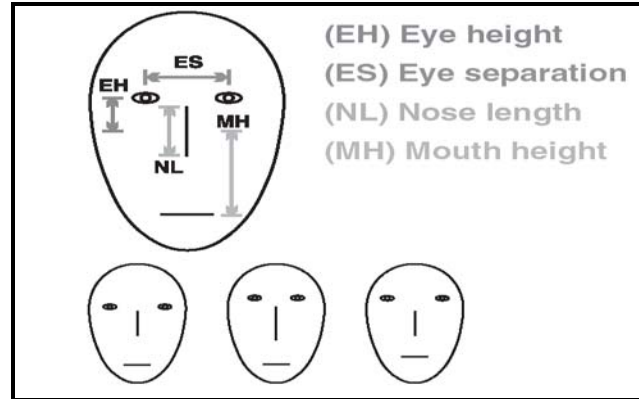


Figure 1. From R.J. Peters et al. / Vision Research 43 (2003) 2265–2280. Features and examples of three different faces from the Brunswik faces task.

the model was trained until it met criterion performance (once through with perfect performance) and then tested on all 16 faces. Building the RPD model consisted of specifying the four binary cues and the possible two courses of action, “Categorize as A” and “Categorize as B”. The RPD model learned the mapping from cues to categories on the fly, without any deeper representation of the problem space, cues, or categories.

The results of the set training condition were that the model performed perfectly on the nine training faces and near-perfectly on the seven generalization faces. This was unlike human performance because the model learned better than people do. However, the model that stopped training after it reached criterion performance more closely matched human performance. This model was a useful early demonstration of the viability of MS/RPD approach because it showed emergent behavior that reflected variability characteristic of human performance on the task, with a very simple representation of cues and courses of action.

More recently, we modeled another categorization task, the more complex Weather Prediction task (described in Santamaria & Warwick, 2008). The Weather Prediction task is a probabilistic category learning task that has been used to study implicit learning. As in the Brunswik faces task, participants predict a binary outcome (rainy or fine weather) based on four binary cues (cards which can be present or absent). However, the 14 different patterns (combinations of cards) that participants see only probabilistically predict the outcomes.

To construct the RPD model, four cues were specified (four cards, each present or absent), and two courses of action were specified (predict rainy and predict fine). Using the distribution from Lagnado et al. (2006) (shown in Table 1), we presented the model with 200 patterns in random order (200 trials). After each pattern, the model

made a prediction and was reinforced according to the probability structure.

Table 1. Frequencies and probabilities of 14 patterns presented to the model of the Weather Prediction task. (Table from Santamaria & Warwick, 2008.)

Pattern	Cues (cards present)				Frequency	Probability of fine weather
	1	2	3	4		
1	1	2	3		19	0.895
2	1	2		4	9	0.778
3	1	2			26	0.923
4	1		3	4	9	0.222
5	1		3		12	0.833
6	1			4	6	0.500
7	1				19	0.895
8		2	3	4	19	0.105
9		2	3		6	0.500
10		2		4	12	0.167
11		2			9	0.556
12			3	4	26	0.077
13			3		9	0.444
14				4	19	0.105
Total					200	0.500

The model learned the probabilistic structure, approaching optimal performance by the third 50-trial block. When examining performance for each pattern separately, we found that the model was sensitive to differences across the 14 patterns. The u-shaped function in Figure 2 reveals that the model picked up on the statistics of the environment, performing at or near chance on weakly predictive patterns, better on moderately predictive patterns, and well on strongly predictive patterns. The model appeared to learn conjunctive information (patterns) rather than elemental information (individual cues).

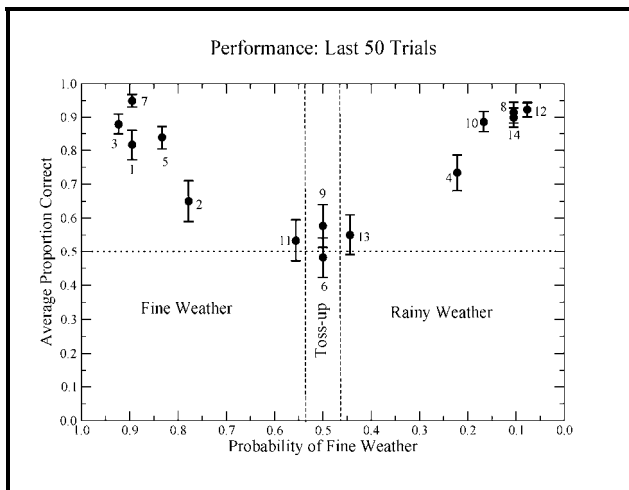


Figure 2. Model performance on the last 50 trials of the Weather Prediction task, averaged across 30 runs, separated by pattern. Error bars are SEM. (Figure from Santamaria & Warwick, 2008.)

These two categorization models invite comparison because they use the same mechanism and have the same structure within the RPD model (four binary cues and two courses of action); they are isomorphic. Despite these similarities, they model two very different kinds of problems that people solve. In the Brunswik faces task, there is an existing partition between categories and a related cue structure to learn. The participant or model must learn where the partition is and then generalize this knowledge to new exemplars. In the Weather Prediction task, there is no transfer/generalization; instead, the participant or model must learn on the fly, in a noisy environment where cues are not perfect predictors.

Categorization models are common tests of modeling approaches and algorithms, and they were a good place to start validating our approach. However, we wanted to push our simple mechanisms to model more complex, unpredictable, even interactive behaviors. To do this, we explored modeling several tasks we see as “dynamic”.

3.2 Dynamic Models

The next three tasks we will discuss can be described as dynamic tasks because the model must adapt to changes in the environment that occur over time. The first and most basic dynamic task we modeled was a very simple vigilance task (presented in Warwick & Hutchins, 2004). Monitoring behavior in a vigilance task is usually represented in a task network model abstractly, as a task of a given duration. In this model, however, we wanted to explore whether appropriate monitoring behavior could be learned from experience.

First we built a periodic activity in the Micro Saint model of the environment. The RPD model had to learn to recognize when this activity was on and decide to attend to it and ignore it when the activity was off, so the two courses of action were “attend” and “ignore”. We tried several cue structures, but the most effective consisted of two cues, one representing the “subjective” passage of time, and the other representing the time since the activity had been observed. As the model ran, it was reinforced for attending when the activity was on and ignoring when the activity was off.

To assess how well the model became attuned to the temporal pattern of activity, we examined two measures: periodicity and weighted error. Periodicity was defined as how frequently the model switched between checking and ignoring activity. Periodicity for a block of time is the number of decisions in that block divided by the number of switches between checking and ignoring. If the model is perfectly attuned to the periodic activity in the environment, the period of its checking behavior should match the period of the activity. However,

periodicity misses the model's alignment with the activity, so we also calculated a measure of weighted error to capture the degree of offset. A small offset would contain mostly errors that occur near a switch (activity on to activity off, or vice versa). These are weighted less than errors that indicate a larger offset.

The model showed evidence of human-like learning: periodicity increased with time to just approach the frequency of the activity. When we set the model to low confidence, a parameter that delays the start of learning so that a small memory of episodes builds up before reinforcement kicks in, weighted error decreased over time, showing improved alignment with the periodic activity (see Figure 3).

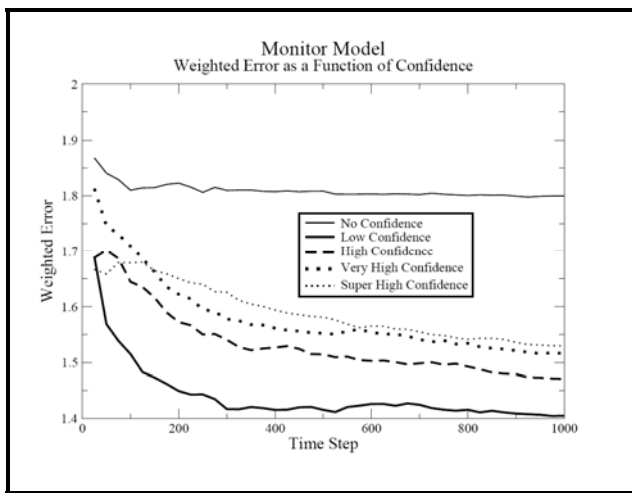


Figure 3. Weighted error in the Monitoring model. For the low confidence condition, weighted error decreases over time, showing that, with experience, the model adapts to the periodic activity.

In later work, we modeled a more richly dynamic task, the Prisoner's Dilemma task, which has been used to study adversarial behavior (presented in Santamaria & Warwick, 2007). This task seems simple, but because it is comprised of two players who react to each other's actions, it can generate complex effects.

The Prisoner's Dilemma is classically presented as two people who are arrested for a crime. Each is offered a deal to testify against the other: if the partner stays silent, the one who testifies goes free and the partner gets the maximum sentence. If both partners testify, they both get a reduced sentence, and if both stay silent, both get a minimum sentence. Unable to communicate about their choice, each person must decide what to do based on what they expect the other person to do. Optimally, both will stay silent. However, as an individual who does not know what your partner will do, the safest move is to testify. A

more informed strategy can be formed when the game is iterated.

In our model, we represented the Prisoner's Dilemma as an abstract 2x2 game, with the payoff structure shown in Table 2. The two players were represented by two separate RPD models which interacted with each other in the Micro Saint Sharp task network environment. Cues for each model were the past n moves for that player and its opponent. The two possible courses of action were cooperate and defect, and success was defined either as "match" (payoff greater or equal than opponent's payoff) or "beat" (payoff greater than opponent's payoff).

Table 2. Payoff structure for the Prisoner's Dilemma.

	<i>A Cooperate</i>	<i>A Defect</i>
<i>B Cooperate</i>	A Payoff=3 B Payoff =3	A Payoff=4 B Payoff =0
<i>B Defect</i>	A Payoff =0 B Payoff=4	A Payoff=1 B Payoff=1

Optimal strategy for this task actually depends on how success is defined. For the "match" condition, the optimal strategy is to always defect, while for the "beat" condition, the optimal strategy is to switch between defecting and cooperating. With six cues (the three most recent moves for each player), the model was able to learn from the opponent's behavior to select the appropriate strategy. In addition, model performance was hurt by reducing memory (having cues for fewer previous moves). Figure 4 shows results for the "match" condition for a model with cues for 1, 2, or 3 previous moves for the opponent. Performance (cumulative payoff) was poorer the more limited the memory.

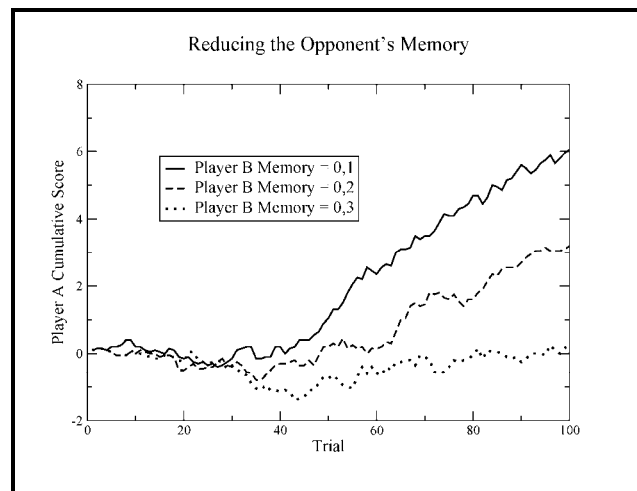


Figure 4. In the Prisoner's Dilemma model, performance is poorer when memory for the opponent's move history goes from 3 previous moves to 2 to 1. (Figure from Santamaria & Warwick, 2007.)

A third dynamic task that we have recently modeled is the Dynamic Stocks and Flows (DSF) task, a classic control task (presented in Lebiere, Gonzalez, & Warwick, in press). The DSF task was developed by Gonzalez & Dutt (2007) to investigate how people perform in dynamically complex environments. It consists of a single stock whose rate of accumulation changes over time. Inflows increase the stock level, and outflows decrease the stock level. Participants try to keep the stock within a certain range. The stock level is changed both by external changes (inflows and outflows) and by user-controlled changes. Controlling the DSF is very difficult for most people: learning tends to be slow and sub-optimal even when participants perform a simplified version of the task that only varies inflow and not outflow.

In building our model of the DSF, we defined cues as the degree of difference from desired stock (relative cues) and courses of action as proportional adjustments to the current stock. Decisions were reinforced whenever the total stock was moved closer to the target stock (rather than an absolute evaluation such as falling within a specified window of the target).

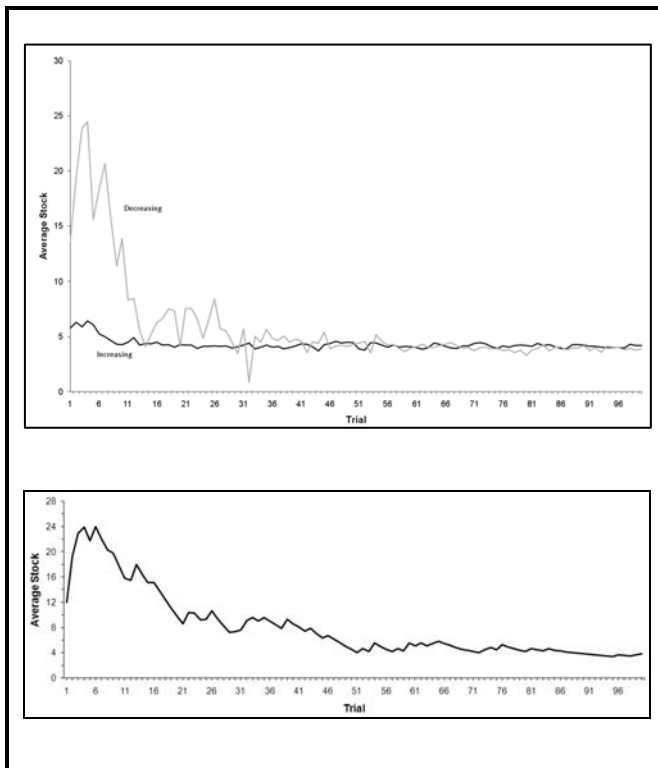


Figure 5. Human (above) and model (below) performance on the Dynamic Stocks and Flows task. The top graph shows both decreasing and increasing flows, and the bottom graph shows decreasing flows.

The model learned more slowly than humans and made more errors, but overall, the model exhibited plausibly

human-like performance (see Figure 5). It seemed to do a much better job controlling the stock in the decreasing condition than in the increasing condition. This seems to happen because in the decreasing inflow conditions, new inflows change the stock less and less and so whatever association was successful before is likely to be successful later.

The Prisoner's Dilemma model and the DSF model are dynamic in the same way; however, their internal representations (in the RPD model) are quite different. The DSF model is purely reactive; it does not explicitly represent the recent past in its cue structure but instead uses a simple delta (difference). In contrast, in the model of the Prisoner's Dilemma, it is critical to represent the recent past; the model shows degraded performance when it can only remember back 1 or 2 moves instead of 3 moves.

The comparison of these two models provides an interesting foil to the comparison of the two categorization models. The two categorization models used the same decision structure but differed in the behaviors they represented, while these two dynamic models capture similar behaviors but require different decision representations.

3.3 A Larger-Scale Model

In addition to modeling categorization tasks and several types of dynamic tasks, we have embarked on some more exploratory efforts that have challenged our modeling approach to go in new directions.

An important exploratory effort was to construct a larger-scale model of a more complex environment. This was not another model validation, but a model comparison effort with a goal of demonstrating face validity for our approach in a more complex, realistic, militarily-relevant environment. The 3 Block Challenge task is a human-in-the-loop simulation of command and control performance in an urban environment developed by Yen and colleagues (see Fan et al., 2006). The three blocks refer to the need to balance three contexts: humanitarian, peacemaking, and combat missions.

Several threats can appear in the 3 Block Challenge: crowds, insurgents, and improvised explosive devices (IEDs). This is a dynamic and complex task environment where threats appear, disappear, and move, available resources are constantly changing, and mission progress and success are dynamic variables. While there are many decisions and complex behaviors that can be modeled in this task, we chose to model the decision of how to allocate resources (friendly units) to the existing threats.

We built a constructive simulation of the 3 Block Challenge in Micro Saint Sharp and built our own RPD model of the resource allocation decision. Yen and colleagues also built a decision model to interact with the Micro Saint Sharp environment, based on their neural-network and agent architecture (also inspired by Klein's recognition-primed decision theory). The goal of this research effort was to compare the two efforts and identify commonalities and differences in assumptions, methods, and approaches.

Because of the online nature of learning in our approach, we iterated over a simple decision: for a given target, for a given unit, should the friendly unit be assigned to the target? The courses of action were then "assign" and "don't assign". The cues given to the model included distance to target, number of units already assigned to the current target, and whether the unit was already assigned to a target.

The model was reinforced when it assigned a unit and 1) the distance from the unit to the target was less than half the maximum screen distance, 2) fewer than two units had been assigned to the target, and 3) the unit was not already assigned. This was a simplified measure of success to evaluate whether the model was exhibiting meaningful behavior and appropriately picking up on the cues in its environment. The model was able to pick up on the predictive value of both individual cues and combinations of cues. Figure 6 shows cumulative success as a function of trial, over 330 decisions, for two representative runs of the model, where the model was given three cues: distance to target, number of units assigned, and whether the unit was already assigned. In both runs, the model increased success over time, and because success was defined in terms of all three cues, this demonstrates that the model was responsive to the combination of three cues.

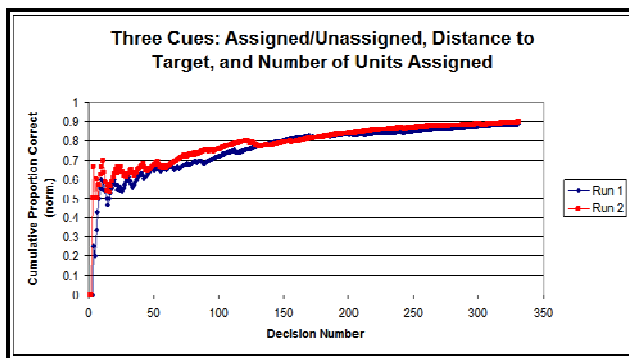


Figure 6. Performance for the 3 Block Challenge model with a combination of three cues.

In comparing this model to its counterpart, developed by our collaborators, we learned that comparison forces all

participating in the comparison to really understand the details of how the different models work, revealing similarities and differences. Similarities included learning from cues to form a representation of a typical situation, and "fuzzy" learning mechanisms (ones that don't rely exclusively on rules and conditionals). Differences included how the decision was framed, how cues were used and integrated, and when the learning occurred (online vs. offline).

The common thread we discovered was the use of fuzzy learning mechanisms and a resulting focus on emergence. Emergence, defined as a high level phenomenon that cannot be explained solely by breaking something down into its components, is a powerful concept that encapsulates key ideas in the RPD theory. This theme, discovered at the implementational level, also informs the high-level theory, an important contribution of model comparison efforts.

4. Discussion

Developing the MS/RPD modeling architecture has been a challenging and thought-provoking endeavor, leading to a closer examination of our assumptions about building models, evaluating them, comparing them, and putting them in context. It has led to new insights with every model, from the practical to the philosophical. Here we will discuss several lessons we have learned from our multi-year modeling effort and then describe our next steps for MS/RPD.

The first lesson we learned is that we can get a lot of complex behaviors and functionality without rich semantic representations, elaborate rules or strategies. Our models are simple, reactive, and use "syntactic" representations that are basically ones and zeros and do not have any built-in meaning. This ability to generate emergent behavior from simple representations is most clearly revealed in the case of the two categorization tasks, where the same internal representation generated two very different kinds of human performance behaviors. However, the choice of representation does matter because everything changes when you try to model dynamic behaviors. Two dynamic tasks we modeled, the Prisoner's Dilemma and the Dynamic Stocks and Flows task, seemed similar but ended up requiring different elements in their internal representations. This is why it was important to go beyond straightforward categorization tasks to test and validate our approach.

Figure 7 shows where the models discussed in this paper fall on two dimensions: the complexity of the task/environment and the complexity of the internal representations used. The simple monitoring model is low in both kinds of complexity, while the scaled-up 3

Block challenge model is high in both kinds of complexity. The two categorization models, the Brunswik Faces model and the Weather Prediction model, have the same internal representations but differ in task complexity. The two dynamic models, The Prisoner's Dilemma model and the DSF model, have similar task complexity but differ in the complexity of their internal representations.

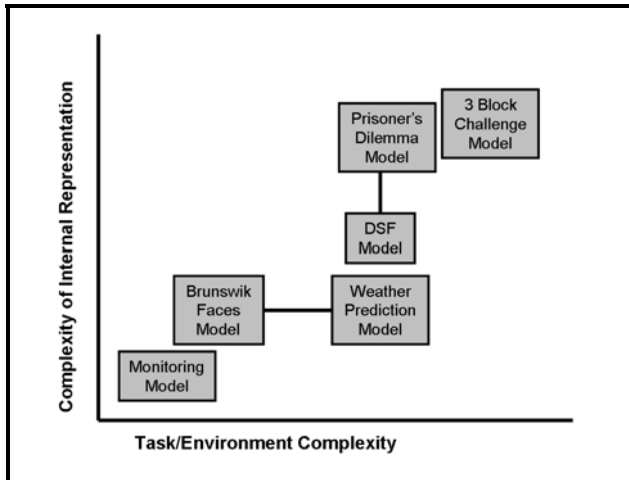


Figure 7. Several MS/RPD models compared on two dimensions: task complexity and complexity of internal representation.

The multiple-trace memory/simple reinforcement mechanism that we employ has shown to be remarkably flexible, capturing a variety of tasks and internal structures and strategies. Because of its ability to pick up on the statistics of the environment, it has been able to model both RPD-like behaviors and more general learning tasks. A criticism we have seen for some of our models is that they do not appear to be particularly “naturalistic” tasks. But we see it as a strength that our mechanisms can be harnessed for a variety of problems as well as more typical recognition-based decision tasks.

The second lesson we learned is that model comparison is an incredibly useful endeavor. A common vocabulary often gets in the way when trying to make side-by-side comparisons of different modeling approaches. By actually building different models of the same task in the 3 Block Challenge (and also the DSF task, comparison effort not discussed here), and comparing mechanisms along with results, we have learned a lot about what different approaches have in common and how they really differ in terms of representations and assumptions. Despite focusing at different levels of abstraction, two approaches may have surprising commonalities, which may result from decisions, such as how to represent a

problem, that occur early in the modeling process. For example, in the 3 Block Challenge model comparison, the use of fuzzy learning mechanisms led to emergent behaviors in both models, but the models differed in the timing of learning (online vs. offline), changing the structure and representation of the decision being modeled.

The third lesson we learned that it is important to go beyond goodness-of-fit criteria when evaluating models and modeling approaches. In particular, it is important to assess the complexity of the model or approach. Models built in the MS/RPD architecture have very few free parameters and require very little tweaking – they work pretty well right out of the box. In addition, we have modeled a fairly wide range of tasks and data with this architecture. Across all of our efforts, we have searched for face validity, looking for patterns of emergent, flexible, human-like behavior. A too-narrow focus on goodness-of-fit might have bogged us down, constrained the tasks we explored, or reduced exploration of model behavior.

In future work with MS/RPD, we will apply the framework to new problems and further explore some of the less used functionalities, such as differential cue weighting and expectancy generation. We will also investigate new functions and uses for the model, such as using the similarity mechanism for pattern completion. We will also systematically investigate the computational limits of our algorithms, scaling up a simple model by adding cues and courses of actions. Finally, we plan to extend MS/RPD to a wider user community by providing more thorough documentation and user support.

5. Conclusions

We have described our MS/RPD approach to modeling decision making in task network models. The goal of this multi-year modeling effort was to increase the cognitive fidelity of task network models. In this paper, we have presented a collection of models and validation results that include classic categorization and probabilistic category learning tasks, dynamic adversarial and control tasks, and a larger-scaled militarily-relevant task, all of which showed human-like behaviors that improve the existing functionality and fidelity of task network models. Using the same simple mechanisms to model this range of behaviors demonstrates the flexibility and utility of our approach. In the future, we will investigate new problems, functions, and applications, we will scale up models to test computational limits, and we will open up MS/RPD to a wider user community.

6. References

- Fan, X., Sun, B., Sun, S., McNeese, M., and Yen, J. (2006). RPD-enabled agents teaming with humans for multi-context decision making. In AAMAS. Hakodate, Hokkaido, Japan.
- Gluck, K. A., Staszewski, J. J., Richman, H. Simon, H. A., and Delahanty, P. (2001). The right tool for the job: Information-processing analysis in categorization. Proceedings for the 23rd Annual Meeting of the Cognitive Science Society, Edinburgh, Scotland.
- Gonzalez, C., & Dutt, V. (2007). Learning to control a dynamic task: A system dynamics cognitive model of the slope effect. Paper presented at the 8th International Conference on Cognitive Modeling, Ann Arbor, MI.
- Hintzman, D. L. (1984). MINERVA 2: A simulation model of human memory. *Behavior Research Methods, Instruments & Computers*, 16, 96-101.
- Hintzman, D. L. (1986a). *Judgments of Frequency and Recognition Memory in a Multiple-Trace Memory Model*. Eugene, OR: Institute of Cognitive and Decision Sciences.
- Hintzman, D. L. (1986b). "Schema Abstraction" in a Multiple-Trace Memory Model. *Psychological Review*, 93, 411-428.
- Klein, G. (1998). *Sources of Power: How People Make Decisions*. Cambridge, MA: The MIT Press.
- Lagnado, D. A., Newell, B. R., Kahan, S., & Shanks, D. R. (2006). Insight and strategy in multiple-cue learning. *Journal of Experimental Psychology: General*, 135, 162-183.
- Lebiere, C., Gonzalez, C., and Warwick, W. (In Press). Emergent complexity in a dynamic control task: A qualitative model comparison.
- Santamaria, A. & Warwick, W. (2007). A naturalistic approach to adversarial behavior: Modeling the prisoner's dilemma. In *Proceedings of the 16th Conference on Behavioral Representations in Modeling and Simulation*.
- Santamaria, A. & Warwick, W. (2008). Modeling probabilistic category learning in a task network model. In *Proceedings of the 17th Conference on Behavioral Representations in Modeling and Simulation*.
- Smith, J. D. and Minda, J. P. (2000). Thirty categorization results in search of a model. *Journal of Experiment Psychology: Learning, Memory, and Cognition*, 26, 3-27.
- Warwick, W. & Fleetwood, M. (2006). A bad Hempel day: The decoupling of explanation and prediction in computational cognitive modeling. In *Proceedings of the Fall 2006 Simulation Interoperability Workshop. Orlando, FL. SISO*.
- Warwick, W. & Hutchins, S. (2004). Initial comparisons between a "naturalistic" model of decision making and human performance data. In *Proceedings of the 13th Conference on Behavior Representation in Modeling and Simulation*.
- Warwick, W., McIlwaine, S., Hutton, R. J. B., & McDermott, P. (2001). Developing computational models of recognition-primed decision making. In *Proceedings of the 10th Conference on Computer Generated Forces*.
- Warwick, W. & Santamaria, A. (2006). Giving up vindication in favor of application: Developing cognitively-inspired widgets for human performance modeling tools. *Proceedings of the 7th International Conference on Cognitive Modeling*.

Author Biographies

AMY SANTAMARIA is a Senior Cognitive Scientist at Alion Science and Technology. Her research focuses on modeling human behavior and cognition and experimentation for robotics interfaces. She received her Ph.D. in Cognitive Psychology and Neuroscience and an M.A. in Cognitive Psychology from the University of Colorado Boulder.

WALTER WARWICK is a Adv. Systems Analyst at Alion Science and Technology. He is working on several projects having to do with the modeling and simulation of human behavior. He received his Ph.D. in History and Philosophy of Science, an Area Certificate in Pure and Applied Logic, and an M.S. in Computer Science from Indiana University.