Metacognition in the Prisoner's Dilemma

Stevens, Christopher; Taatgen, Niels; Cnossen, Fokeltje

Published in:
proceedings of the 13th Annual International Conference on Cognitive Modeling

Metacognition in the Prisoner's Dilemma

Christopher A. Stevens (c.a.stevens@rug.nl), Niels A. Taatgen (n.a.taatgen@rug.nl), Fokie Cnossen (f.cnossen@rug.nl)
Department of Artificial Intelligence, Nijenborgh 9
9747 AG Groningen, The Netherlands

Abstract

In this paper, we show ACT-R agents capable of metacognitive reasoning about opponents in the repeated prisoner's dilemma. Two types of metacognitive agent were developed and compared to a non-metacognitive agent and two fixed-strategy agents. The first type of metacognitive agent (opponent-perspective) takes the perspective of the opponent to anticipate the opponent's future actions and respond accordingly. The other metacognitive agent (modeler) predicts the opponent's next move based on the previous moves of the agent and the opponent. The modeler agent achieves better individual outcomes than a non-metacognitive agent and is more successful at encouraging cooperation. The opponent perspective agent, by contrast, fails to achieve these outcomes because it lacks important information about the opponent. These simple agents provide insights regarding modeling of metacognition in more complex tasks.

Keywords: Theory-of-mind; Metacognition; Prisoner's Dilemma; ACT-R

Metacognition in Two-Person Games

Humans can reason about the minds of others and predict their behaviors, a metacognitive ability known as theory of mind (Premack & Woodruff, 1978). A question of great current interest is why humans evolved this ability. One possible reason is that theory of mind allows people to understand and predict the actions of others (McCabe, et al., 2000). People can use this ability to determine whether another person is likely to be cooperative or competitive. Theory of mind might also be used to learn the strategy of an opponent and devise an appropriate counter-strategy (Hingston et al., 2007).

The prisoner's dilemma is a task that embodies the basic conflict between cooperation and competition often found in real-world interactions. It is often used to study how various strategies may help or harm an individual's or group's chances of survival (Axelrod, 1980; Wedekind & Melinski, 1996; Nowak & Sigmund, 1993). However, very little is known about how metacognition impacts performance in this task. In the present work, we develop two cognitive agents that embody different metacognitive strategies. We then compare the performance of these agents against an existing, non-metacognitive agent (Lebiere, Wallach, & West, 2000) and two normative strategies (tit-for-tat: Axelrod, 1980; win-stay-lose-shift: Wedekind & Melinski, 1996).

The Prisoner's Dilemma

The prisoner's dilemma is a 2 x 2 game in which players must choose to cooperate with their opponent (move B) or to defect (move A). This results in one of four possible outcomes. The following is a typical payoff matrix for the prisoner's dilemma game.

		Player 2	
		Cooperate (B)	Defect (A)
Player 1	Cooperate (B)	1,1	-10, 10
	Defect (A)	10, -10	-1, -1

If both players consistently choose to cooperate, then they both will enjoy a positive payoff. However, cooperation is risky, because both players have a temptation to defect. If the opponent defects when a player cooperates, the cooperating player will lose a large number of points. Defection also carries risks. When there is more than one round, opponents may retaliate by defecting in later rounds. The optimal strategy is not obvious and depends on the opponent. Therefore, reasoning about an opponent's goals and predicting their future behavior should provide an advantage (Hingston et al., 2007). To determine if this is true, we developed two cognitive agents that represent different strategies for metacognitive reasoning. We then tested these agents against fixed strategies and a non-metacognitive model.

The ability to reason about others may have implications not only for individual outcomes, but also for collective outcomes. De Weerd, Verbrugge, and Verheij (2013) developed agents with different levels of metacognitive ability and pitted them against one another in a negotiation game. They found that agents that could reason about their opponents' beliefs obtained both greater rewards for themselves and found opportunities for greater collective rewards. In a similar way, metacognitive abilities may improve cooperation in the prisoner's dilemma. Metacognitive agents may have an easier time predicting their opponents, allowing them to know when cooperation is possible.

To evaluate the metacognitive agents presented here, we used a previous agent developed by Lebiere, Wallach, & West (2000) as a baseline. This agent is built within the ACT-R cognitive architecture (Anderson et al., 2004). The agent provides a good fit to human data, but it is not metacognitive because it bases its decisions only on the immediate payoffs of its previous moves. It does not

attempt to learn its opponent's strategy or explicitly reason about its opponent. For this reason, we hereafter refer to it as the self-payoff agent.

We present two types of metacognitive agent: opponent-perspective and modeler. The opponent-perspective agent is inspired by the simulation theory of mind (Gallese & Goldman, 1998; Meltzoff, 2007), which states that people reason about the mental states of another by adopting the other's perspective. On every trial, the agent computes the move that it would make if it was the opponent and then selects its own move accordingly.

The modeler agent is inspired by opponent-modeling agents in the multi-agent systems literature (e.g. Hingston et al., 2007) as well as models of sequence learning in ACT-R (Lebiere & West, 1999). An opponent-modeler agent develops a mental model of the opponent's strategy over time and attempts to predict the opponent's next move probabilistically. Hingston et al. (2007) present an opponent-modeler that can successfully play the prisoner's dilemma against a variety of other agents. However, this agent is not a cognitive agent, and does not attempt to capture the flexibility or variability of human behavior. Our modeler agent extends this approach by using the declarative memory system of ACT-R to create a cognitively plausible, dynamic model of its opponent that can be rapidly updated to handle new information. This approach should be helpful for adapting to new opponents or strategy shifts in current opponents.

Simulations

In the following simulations, we use an instance-based learning approach to allow the agents to adapt to their opponents (Logan, 1988). In this approach, the outcomes of previous trials are encoded as chunks in declarative memory. The agents then attempt to predict outcomes or opponent behaviors by retrieving a chunk from memory that matches the current situation. By updating the contents of declarative memory, the agents can adapt their strategies to suit their opponent.

The likelihood of retrieval of a chunk in ACT-R is determined by its activation level. The more frequently and recently a chunk has been used, the more active it will be. For all simulations, we used the full (non-optimized) learning equation of ACT-R:

$$B_i = \ln\left(\sum_{j=1}^{n} t_j^{-d}\right) + Logistic(0, s)$$

In this equation, n is the number of presentations of chunk i. t_j is the time since the jth presentation. A presentation is either the creation of a chunk or a retrieval of that chunk. d is the rate of activation decay. By default this is set to 0.5. The rightmost term of the equation represents noise added to the activation level. s is an ACT-R parameter that determines the standard deviation of the noise. For all simulations reported here, we use an s value of 0.25, consistent with the value used in Lebiere, et al.'s (2000) model.

In the following simulations, we compare the performance of the two metacognitive agents to the self-payoff agent. The aim was to determine whether metacognitive reasoning can give an agent more robust performance across a variety of agents. To do this, we played all three of these agents against the self-payoff agent and two fixed-strategy agents. We hypothesized that the metacognitive agents would have better overall performance across all opponents because of their greater adaptability.

The strategies of the fixed-strategy agents were based on two normative strategies found in the prisoner's dilemma literature: tit-for-tat (TFT; Axelrod, 1980) and win-stay-lose-shift (WSLS) (Nowak & Sigmund, 1993). Both of these strategies have been shown to provide robust performance against a variety of opponents. There are several variations of the tit-for-tat strategy, but all of the variations tend to copy the previous move of their opponent. We used a strict TFT strategy that always copied the previous move of the opponent. The WSLS strategy, by contrast, continues to make the same move until it loses points, then it changes moves.

The Self-Payoff Agent

This agent is a replication of the model reported in Lebiere, et al. (2000). It was originally designed to account for behavior in the prisoner's dilemma task without resorting to notions of altruism or to long-term payoff calculations. It does not attempt to explicitly reason about its opponent or predict its opponent's behaviors. Instead, it predicts the most likely payoff of each of its possible moves. Then it selects the move associated with the highest payoff.

The self-payoff agent remembers the previous rounds of the game using four declarative memory chunks. Each chunk represents one of the four possible outcomes of the game.

```
A1-A2 isa outcome move1 A move2 A payoff1 -1 payoff2 -1
A1-B2 isa outcome move1 A move2 B payoff1 10 payoff2 -10
B1-A2 isa outcome move1 B move2 A payoff1 -10 payoff2 10
B1-B2 isa outcome move1 B move2 B payoff1 1 payoff2 1
```

The four outcomes are A1A2, A1B2, B1A2, and B1B2. The first letter of the pair represents player 1's move and the second letter represents player 2's move. The move1 and move2 slots represent the moves chosen by players 1 and 2 respectively. The payoff slots contain the number of points both players receive. In every round, the model creates a new outcome chunk in the goal buffer. When the model selects its move, it records it in the move1 slot. At the end of the trial, the opponent's move and the resulting payoffs are also recorded in this chunk.

The self-payoff model uses the relative activation levels of these four chunks to determine the most likely outcome of a given move. It does this by using a set of four

production rules. The first two productions retrieve two outcome chunks, one in which move1 is A and one in which move1 is B. The remaining productions then select move A if payoff A is higher or move B is payoff B is higher.

The self-payoff agent provides a good fit to human data both in the prisoner's dilemma and in other 2 x 2 games (Lebiere et al., 2000). It can account for cooperative behavior because the A1-A2 and B1-B2 chunks should become more active over time (as long as the opponent is not a consistent defector). When these chunks are more active than the other two chunks, the agent determines that cooperation is more profitable than defection.

The Opponent-Perspective Agent

The opponent-perspective agent is an adaptation of the Lebiere et al. (2000) model that attempts to predict the opponent's move by deciding which move it would take in the opponent's place. The opponent-perspective agent stores the same information in memory as the self-payoff agents. But instead of determining its own most likely outcomes, it predicts the most likely outcomes for its opponent. It then predicts that its opponent will select the move with the highest payoff. Based on this prediction, it selects an appropriate response.

Due to the nature of the prisoner's dilemma, there is not an optimal countermove for each possible opponent move. Regardless of an opponent's move, defecting will always lead to a higher immediate payoff than cooperating. Therefore, we designed these agents to use an imitative strategy. That is, the agent will do what it thinks the opponent is going to do this round. If the opponent's behavior can be successfully predicted, then the agent will be able to find opportunities for cooperation without being exploited.

The perspective model uses the same declarative chunks and production rules described above. However, its production rules instead compare the opponent's payoffs rather than its own payoffs. Based on this comparison, the model will predict the opponent's next move and select the same one.

In principle, this agent should be able to perform well against the self-payoff agent and the normative strategies. The metacognitive agent uses the same payoff calculation as the self-payoff agent, and this should make it easier to predict the self-payoff agent's moves. The TFT and WSLS agents calculate their moves differently, but they are all based on the same principle of cautious cooperation. When the opponent cooperates and punishes defection, these agents will tend to cooperate more.

The Modeler Agent

The modeler agent represents a different form of metacognition than the opponent-perspective agents. The modeler attempts to build a mental model in declarative memory to predict the opponent's most likely next move based upon their previous moves. Unlike Hingston et al.'s (2007) opponent-modeler, the modeler makes a specific prediction about the move the opponent is going to make in the current round. Also, because it makes use of ACT-R's declarative memory system, the modeler weighs information from more recent rounds more heavily than less recent rounds. This should afford the agent greater flexibility in its behavior.

Unlike the opponent-perspective agent, the modeler does not make any assumptions about the specific strategy used by the opponent. Instead, it tracks how the opponent responds to each of the four possible outcomes in the game (double-defect, defect-cooperate, cooperate-defect, and double-cooperate). It then predicts that the opponent will make the same move after the outcome appears again.

The memory structure of the modeler agent is different from that of the self-payoff and opponent-perspective agents. The model does not start with any predefined chunks, but after every trial, it will create a new chunk like the following example:

SEQUENCE0 isa sequence move1 A move2 B next-move A

Move1 represents the player's move and move2 represents the opponent's move. Next-move represents the opponent's move on the following round. This chunk represents an instance in which the agent defected and the opponent cooperated; in the next round, the opponent responded with a defection.

Before deciding on a move, the modeler agent will retrieve a previous instance that matches the current situation. For example, after a double-cooperation round, the modeler will attempt to retrieve a chunk in which both move1 and move2 are B. Based on this retrieved chunk, it will predict the opponent's next move. Like the opponent-perspective agent, the modeler will select the same move that it thinks its opponent is going to select. If this prediction turns out to be incorrect, the modeler will create a new chunk to reflect the correct prediction and store it in memory. If the modeler fails to retrieve a similar instance, it will select a move randomly.

Simulation Results

Fifteen simulations were run. The self-payoff, opponent-perspective, and modeler agents were all played against all other agents. Each simulation consisted of 1000 runs of 100 trials. Results were averaged over the runs. A summary of the performance can be found in Tables 1 and 2.

Self-payoff

The self-payoff agent behaved consistently with the version previously reported (Lebiere et al., 2000). When the self-payoff agent plays against itself, some runs are strongly cooperative (A1A2 = 4%; B1B2 = 92%) and in others there is no cooperation at all (A1A2 = 96%; B1B2 = 0%). However, when all of the runs were averaged together, the

Table 1. Individual Scores of Agent 1 (95% confidence intervals in parentheses)

Agent 1	Agent 2				
	Self-payoff	Opponent – perspective	Modeler	TFT	WSLS
Self-payoff	-56 (±7)	27 (±13)	-19 (±6)	-58 (±3)	250 (±7)
Opponent-perspective	-147 (±11)	-37 (±22)	-67 (±4)	-56 (±3)	335 (±11)
Modeler	-46 (±4)	-61(±4)	-1 (±5)	9 (±6)	103 (±1)

self-payoff agent demonstrated an overall tendency to play aggressively. Overall, the self-payoff agent defected on 73% of the trials. Given the design of the agent, it may seem peculiar that it has such a strong tendency to defect. Reinforcing the A1A2 chunk should make the cooperate move more appealing (because cooperating may yield +1 rather than -1). The answer to this puzzle may lie in the agent's prediction process. Each time the agent retrieves an outcome, that outcome is reinforced in declarative memory, even if it never occurs. In other words, the expectations of the agent are self-reinforcing. If the agent frequently retrieves the B1A2 chunk by chance, the B1B2 chunk may never have a chance to become sufficiently active. Moreover, cooperation is fragile because two conditions must be met for the model to select cooperate. The model must believe that defection will be punished (A1A2 > A1B2) and that cooperation will not be exploited (B1B2 > B1A2). If the opponent cooperates too frequently, then the agent will attempt to exploit it. If the opponent defects when the agent tries to cooperate, it will quickly retaliate.

Against TFT, the self-payoff agent earned a low negative individual score and combined score. The TFT agent swiftly and consistently punishes defection, but it will only cooperate again after its opponent has cooperated. This results in a loss for the self-payoff agent, making it less likely to cooperate in the future.

The self-payoff agent earned a very high score against WSLS, but it did so by exploiting WSLS's cooperation. The average score of the WSLS agent was very low. This is probably due to the self-payoff agent's strong tendency to defect. When the opponent defects, the WSLS agent loses points and therefore switches strategies. As a result, the WSLS agent essentially became a random decision agent because it lost points regardless of its move. This constant switching made the WSLS agent extremely vulnerable to exploitation.

Opponent-Perspective

The opponent-perspective agent performed the worst of the three agents. In terms of individual outcomes, it earned the lowest score against the self-payoff and modeler agents. It performed the best against the WSLS agent, but only because it tended to exploit the WSLS agent's cooperation.

The self-payoff agent heavily exploited the opponent-perspective agent, often defecting when the opponent-perspective agent cooperated (B1A2 = 12%). This happened mostly on runs in which the self-payoff agent very rarely cooperated. On these runs, the opponent-perspective agent's A1A2 chunk and A1B2 chunk were both highly active (because both outcomes are very frequent). The B1A2 and B1B2 chunks, on the other hand, only receive activation from the internal predictions of the model. This sometimes causes the B1B2 chunk to become highly active, leading the agent to predict cooperation. To make matters worse, when the opponent-perspective model chose to cooperate, it reinforced the A1B2 chunk of the self-payoff agent, reinforcing its tendency to defect. As a result, the rate of mutual cooperation was quite low.

Against the TFT agent, the performance of the opponent-perspective agent was very similar to the self-payoff agent. It showed a slight tendency to exploit the TFT agent (A1B2 = 5%). And, like the self-payoff agent, the rate of mutual cooperation was low. The reason why the opponent-perspective agent is not exploited by the TFT agent is because the TFT agent will always answer a cooperation with a cooperation. So when the TFT agent does unilaterally defect, the opponent-perspective agent has the opportunity to do the same next round.

Like the self-payoff agent, the opponent-perspective agent had a strong tendency to exploit the WSLS agent. The two most common outcomes were mutual defection (A1A2 = 38%) and unilateral defection by the opponent-perspective agent (A1B2 = 38%). The high activation of the A1B2 chunk caused the opponent-perspective agent to predict that the WSLS agent would never cooperate because it lost points so frequently as a result of doing so.

Modeler

The modeler agent, by contrast, did succeed in both achieving more favorable outcomes for itself and learning to cooperate with other agents when possible. The modeler obtained higher scores than both other agents against the self-payoff and the TFT agents. It also demonstrated a high positive score against the WSLS agent without exploiting it.

Table 2: Percentage of Joint Cooperation Trials (B1B2) (95% confidence intervals in parentheses)

Agent 1	Agent 2				
	Self-payoff	Opponent-perspective	Modeler	TFT	WSLS
Self-payoff	13 (±1)	10 (±1)	29 (±2)	11 (±2)	53 (±2)
Opponent-perspective	-	7 (±1)	13 (±2)	13 (±2)	19 (±2)
Modeler	-	-	45 (±2)	51 (±3)	97 (±.1)

The modeler agent does not achieve perfect prediction against the self-payoff agent. In fact, when the modeler plays against the self-payoff agent, the self-payoff agent achieves a higher score. However, the modeler is more successful at encouraging the self-payoff agent to cooperate. This improves the collective score and explains why the modeler scores more points when it plays against the self-payoff agent than the self-payoff agent does when it plays against itself. In addition, the modeler agent is able to predict the self-payoff agent well enough that it can avoid the heavy exploitation suffered by the opponent-perspective agent.

The modeler agent achieves the highest rates of cooperation of all three agents, demonstrating that the agent can quickly learn when cooperation with an opponent is possible. By a large margin, the modeler agent obtains the most mutually cooperative trials. There is room for improvement, however. The joint score of the modeler against the TFT agent is far from the ideal 200 points. This is because early defection from the TFT agent can lead the modeler agent to expect defection and respond in kind. This is not so with the WSLS agent, which changes to cooperation after a mutual defection.

Discussion

The fundamental problem for the player in the prisoner's dilemma is knowing when the opponent can be trusted. Our simulations suggest that metacognitive reasoning, if done appropriately, can help to solve this problem. Only one of the metacognitive agents we tested demonstrated an advantage over a non-metacognitive agent. The modeler agent was successful both in increasing its own individual gains and in discovering opportunities for cooperation with opponents. The opponent-perspective agent, however, was not able to achieve better outcomes for itself or find opportunities for cooperation.

Overall, the modeler agent is the best of the metacognitive agents because of its ability to flexibly adapt to different opponents. Against cooperative agents, it will quickly learn to cooperate and achieve positive scores. Against aggressive agents, it will learn to play defensively and defect most of the time. In addition, he modeler agent demonstrates one example of how metacognition may work to improve collective outcomes as well as individual outcomes. In interactions like those in the prisoner's dilemma, uncertainty can be a major obstacle to cooperation. Metacognitive reasoning may help to make other agents more predictable. When agents can be confident that their partners will cooperate, they may be more willing to cooperate themselves.

However, the modeler agent has several important limitations. One current limitation of the modeler agent is that it represents a very simple theory of mind because it does not represent the opponent's declarative knowledge or beliefs. These were not necessary for the present purposes because of the simplicity of the task, but modeler agents in more complex tasks will likely require such representations. An additional limitation is that it does not consider the relative payoffs of its choices. Rather, it imitates the opponent. This makes sense in the prisoner's dilemma, where unilateral choices (AB and BA) are generally avoided by agents because of severe costs. But it may not extend well to other tasks. This limitation could be addressed by making the model make three predictions: (1) the opponent's move on the current trial and (2) the opponent's response to each of the agent's possible moves. The model could then select the move that will lead to the highest immediate and future payoffs. The same declarative chunks used by the model to predict the opponent's current move could also be used to predict how the opponent will respond to the agent's current move.

The present simulations, together with those of Kennedy and Krueger (2013), highlight important challenges in modeling theory of mind. A "like me" agent (Meltzoff, 2007) may fail if the agent has access only to one strategy or a small number of strategies. This prevents the agent from considering that the opponent may be approaching the task in a different way. Kennedy and Krueger used a "like me" approach to develop a theory-of-mind agent that could play a voluntary trust game. This agent computed that it could achieve the highest average score by defecting. Believing that the opponent would reach the same conclusion, the agent always defected. Our opponent-perspective model shares a similar weakness. Because the short-term payoffs are skewed in favor of defection, both agents predict that their opponents will have a strong tendency to defect. In our simulations, this prevented the opponent-perspective agent from predicting the behavior of the more cooperative agents (TFT and WSLS). These results do not mean that taking the opponent's perspective is not helpful. But it may be necessary for agents to have access to a larger set of strategies so that they can find one that resembles the opponent's behavior. For example, if a model had two strategies (e.g. one cooperative and one aggressive), it could make predictions for the opponent's behavior based on both strategies. It could then select its own strategy based on which one was a better fit to the opponent's behavior.

A further challenge in constructing "like me" agents for 2-person games is tracking stochasticity in an opponent's behavior. Human performance in many of these games contains varying degrees of noise (Lebiere & West, 1999). Even if an agent has access to the same memory structure and decision rules as an opponent, that agent may still have difficulty tracking moment-to-moment variations. The opponent-perspective agent had a difficult time predicting the behavior of the self-payoff agent because it did not have access to its trial-by-trial predictions. On some runs, the self-payoff model's B1A2 chunk became active very early on by a series of chance retrievals, making it very unlikely to cooperate. However, the opponent perspective model did not know this, and still predicted that the self-payoff model would cooperate.

Opponent modeling, as opposed to the "like me," approach, is a more flexible strategy for a theory of mind

agent. Such agents are not limited by their own repertoire of strategies, and can successfully predict a wider range of opponents. In some cases, this approach may also be more cognitively efficient because it does not require mentally simulating the opponent's decision process. These agents may be especially powerful in situations in which opponents can change strategy without warning. If the agent has an adaptive declarative memory system, it could quickly update its mental model of the opponent and counteract the new strategy.

However, opponent modeling is not without drawbacks. It may be harder to implement such a strategy in more complex tasks. In the prisoner's dilemma, there are only two possible behaviors and an opponent's behavior is completely visible. When a greater number of behaviors is possible, it may be more difficult for an agent to determine which opponent behaviors are relevant.

These simulations do not address how well the model behavior replicates human behavior, nor does it show how the models would perform against humans. Playing against humans would provide a much stronger test for the modeler agent, as humans are likely to employ a variety of strategies and change strategies as the game progresses. We are currently planning an experiment in which we will collect this data. Of particular interest here is whether the modeler will be as successful in encouraging humans to cooperate as it is with the TFT and WSLS agents.

The work shown here demonstrates that metacognitive reasoning about an opponent can improve outcomes both for oneself and for one's opponent in the prisoner's dilemma. By learning an opponent's strategy, an agent can determine if it is safe to cooperate, or if it is better to defect. This increases the probability that the agent and its partner will discover a stable, mutually beneficial outcome. Metacognition also helps an agent detect and defend itself against cheaters. However, players should beware to avoid assuming that all opponents will play the game the same way they do. We expect that these benefits extend not only to other simple games but also to more complex scenarios. It remains for further work to discover how metacognitive reasoning can be best employed to achieve success in these other tasks.

Metacognitive reasoning about an opponent's behavior can provide an advantage in the repeated prisoner's dilemma. The ability to predict an opponent's next move helps to determine when it is safe to cooperate and when one should defect. This suggests that performance even in simple games may benefit from developing a theory of mind about one's opponent.

Acknowledgments

This work was funded by European Union Grant #611073: Multiperspective Multimodal Dialogue (METALOGUE).

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036.

Axelrod, R. (1980). More Effective Choice in the Prisoner's Dilemma. *Journal of Conflict Resolution*, *24*(3), 379–403.

de Weerd, H., Verbrugge, R., & Verheij, B. (2013). Higher-order theory of mind in negotiations under incomplete information. In *PRIMA 2013: Principles and Practice of Multi-Agent Systems* (pp 101-16) Springer: Berlin Heidelberg.

Gallese, V., & Goldman, A. (1998). Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, *2*(12), 493–501.

Kennedy, W. G., & Krueger, F. (2013). Building a Cognitive Model of Social Trust Within ACT-R. In *AAAI Spring Symposium: Trust and Autonomous Systems* (pp. 29–34).

Hingston, P., Dyer, D., Barone, L., French, T., & Kendall, G. (2007). Opponent Modelling, Evolution, and the Iterated Prisoner's Dilemma. In *The Iterated Prisoner's Dilemma: Celebrating the 20th Anniversary* (pp. 139–170). World Scientific.

Lebiere, C., Wallach, D., & West, R. (2000). A memory-based account of the prisoner's dilemma and other 2x2 games. *Proceedings of International Conference on Cognitive Modeling*. 185-93.

Lebiere, C., & West, R. L. (1999). A dynamic ACT-R model of simple games. In *Proceedings of the Twenty-first Conference of the Cognitive Science Society*, pp. 296-301. Mahwah, NJ: Erlbaum.

Logan, G.D. (1988). Toward an instance theory of automatization. *Psychological Review, 95*, 492-528. Logan,

McCabe, K. a, Smith, V. L., & LePore, M. (2000). Intentionality detection and "mindreading": why does game form matter? *Proceedings of the National Academy of Sciences of the United States of America*, *97*(8), 4404–9.

Meltzoff A. N. 2007. Imitation and Other Minds: The "Like Me" Hypothesis. In: S. Hurley and N. Chater (eds) Perspectives on Imitation: From Neuroscience to Social Science, pp 55-77. Cambridge: MIT Press.

Nowak, M., & Sigmund, K. (1993). A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, *364*, 56–58.

Premack, D., & Woodruff, G. (1978). Does the chimpanzee have a theory of mind?. *Behavioral and brain sciences*, *1*(04), 515-526.

Wedekind, C., & Milinski, M. (1996). Human cooperation in the simultaneous and the alternating Prisoner's Dilemma: Pavlov versus Generous Tit-for-Tat. *Proceedings of the National Academy of Sciences of the United States of America*, *93*(7), 2686–9.