Programming Agents as a Means of Capturing Self-Strategy[*]

Michal Chalamish
Computer Science Department
Bar-Ilan University
Ramat-Gan 52900, Israel
merlich@cs.biu.ac.il

David Sarne
Computer Science Department
Bar-Ilan University
Ramat-Gan 52900, Israel
sarned@cs.biu.ac.il

Sarit Kraus [†]
Computer Science Department
Bar-Ilan University
Ramat-Gan 52900, Israel
sarit@cs.biu.ac.il

ABSTRACT

In this paper we report results of an extensive evaluation of people's ability to reproduce the strategies they use in simple real-life settings. Having the ability to reliably capture people's strategies in different environments is highly desirable in Multi-Agent Systems (MAS). However, as trivial and daily as these strategies are, the process is not straightforward and people often have a different belief of how they act. We describe our experiments in this area, based on the participation of a pool of subjects in four different games with variable complexity and characteristics. The main measure used for determining the closeness between the two types of strategies used is the level of similarity between the actions taken by the participants and those taken by agents they programmed in identical world states. Our results indicate that generally people have the ability to reproduce their game strategies for the class of games we consider. However, this process should be handled carefully as some individuals tend to exhibit a behavior different from the one they program into their agents. The paper evaluates one possible method for enhancing the process of strategy reproduction.

Categories and Subject Descriptors

I.2.11 [**Computing Methodologies**]: Artificial Intelligence—*Distributed Artificial Intelligence*

General Terms

Human Factors, Experimentation

Keywords

large-scale MAS, strategic behavior, human behavior

1. INTRODUCTION

Modeling individuals and reliably capturing their behavior is a major challenge in AI [12]. In particular, the need to reproduce strategic behavior of individuals is inherent in the design of simulation systems in which the modeled individuals are not-necessarily cooperative or are completely self interested (i.e., having different objectives, or are unreliable [7]). In recent years we have witnessed a substantial increase in the use of autonomous agents in medium and large-scale simulation systems. This is mainly due to the advantage of representing people and complex objects by agents that can interact among themselves and scale [21]. Indeed, the key challenge in these systems remains, as noted by many researchers, the complexity of modeling people's behavior [14]. Here, the use of traditional methods for modeling individual behaviors by domain experts (e.g. [17, 11, 19]) has turned out to be problematic. Problems arise usually in scenarios where the agents are expected to exhibit human behavior, while the strategies embedded in these agents were derived from real data collected under dissimilar settings. Furthermore, it is very difficult to use historical data to capture the different variations exhibited in people's behavior. In particular it is difficult to capture variations resulting from the way people are affected by the behavior of others, as both the environment and the number of other agents they operate alongside change.

Unlike computer agents (that are usually rational and significantly less bounded computationally), people can not be trusted to exhibit the equilibrium or any other logical expected behavior [15]. An example of the above difficulty can be found in the traffic simulation domain where many of the simulation tools that were developed and implemented in recent years have been found to have problems in their accuracy of representing the traffic flow [5]. Similarly, we note the evacuation and disaster recovery domains, where many attempts have been made to develop crowd simulators [13, 6]. Nevertheless, as evidenced in literature, people's behavior in these domains (regardless of being fully or partially self-interested) cannot be fully captured by a simple numerical analysis of inputs such as the positions of people and structures. Instead, one has to recognize that a person's behavior can be affected by a large set of social (e.g., leadership [13]), psychological and physiological [6] factors.

One option to resolve this problem is to extract behaviors directly from people [3, 8, 2]. This unique approach to the problem relies on a Multi-Agent based simulation in which each participating agent is pre-programmed by a different person, in a distributed manner, capturing her strategy to be exhibited in scenarios similar to those that need to be simulated [2]. Thus, a significant number of simulated indi-

[*]This work was supported in part by NSF under grant #IS0705587 and ISF under grant #1357/07

[†]Sarit Kraus is also affiliated with the University of Maryland Institute of Advanced Computer Studies.

Cite as: Programming Agents as a Means of Capturing Self-Strategy, Michal Chalamish, David Sarne and Sarit Kraus, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. XXX-XXX.

viduals can be generated, each equipped with behaviors that reliably clone the behavior of an individual in the simulated system. The advantages of this approach are numerous: the agents' strategy creation can now be distributed to a large group of people and a large set of strategies can be obtained, in a cost-effective manner, sufficiently representing the simulated population, and scaled-up as needed by simply creating more instances of each agent type/strategy.

Nevertheless, while the latter method enables the creation of a wide range of strategies, there is no guarantee that the strategies reported (and programmed) by people are actually a reliable reflection of their "real-life" strategy. In case of a significant deviation between the two, the use of programmed strategies will lead to non-credible results. As we describe in the related work section (Section 6), there are many evidences in the literature for discrepancies between people's actual behavior and the way they report it. Consequently, prior to delegating the task of programming the agents' strategies to people, one needs to attain a good estimation of how accurate people are in extracting their own strategies (and programming them into a computer agent) in the domains under question. Furthermore, if there is a gap between the way people act and their stated strategies, then there is a great incentive to evaluate tools and methodologies which will bridge this gap.

In this paper we attempt to empirically evaluate the ability of people to extract and reproduce their exhibited strategies in a class of games (i.e., environments) with a moderate level of complexity. Specifically, the environments we consider are characterized by a sequence of decision points, each having a limited set of options from which the user may choose, with an outcome depending on the outcomes of former decision points or external events that occurred till the current stage of the game. Still, none of the games under consideration can be solved computationally (in real-time as the game is played) by the person playing it. This class of environments includes many tasks that users often face on a daily basis or are likely to be familiar with the situation and its possible set of actions (e.g., searching to buy a product, searching for a parking space in a parking lot, playing card games, deciding what lines to join at Disneyland, deciding on a route to follow in evacuation scenarios[1]).

For this purpose we designed four different games (differing both in their complexity and their strategy characteristics), representing common tasks that most users have experienced before and are most likely to be familiar with. We compare the actions taken by participants in thousands of game sessions of these types with the actions of agents programmed by the participants for these games. This methodology allows us to empirically learn about the nature of the differences between the strategies people actually use and the strategies they embed into their agents for this class of environments. In addition to using the large-scale evaluation scenario in order to understand how close we are to being able to rely on people to capture their strategies, we attempt to evaluate the usefulness of a possible approach to enhance their ability. The approach suggests an iterative process in which participants are able to see the differences between the performance of the agents they have programmed and the behaviors they have exhibited in the experiments.

The remainder of this paper is organized as follows. Important concepts and definitions relating to measuring the similarity between people and their agents are given in Section 2. The description of the games used in our experiments is detailed in Section 3. Section 4 presents our methodology and experimental settings. The results are analyzed in Section 5. Related work is reviewed in Section 6. We conclude with a discussion and directions for future research (Section 7).

2. STRATEGIES AND DECISION POINTS

As described in the introduction, the games we consider are based on decision points. A *decision point* is a stage in the game where the player is required to choose an action to be taken. We define a person's or an agent's *behavior* as a sequence of actions taken during a single game session.

The term *strategy*, in this context, refers to some general form of action used by a player in a game to achieve her goal (e.g., to win or maximize a score). Formally, let $A = \{a_1, a_2, \cdots, a_n\}$ be the set of actions a person or an agent can take at any decision point in the game and let $DP = \{dp_1, dp_2, \cdots, dp_m\}$ be the set of all the decision points in the game. Strategy S is the function $S : DP \rightarrow A$, which chooses the specific action to be taken at each decision point. Note that A might include actions which are not valid in some of the decision points.

We use the function *Closeness*(Person,Agent) to measure the closeness (i.e., the similarity level) between the behavior displayed by a Person and an Agent. A higher value for closeness means more similarity between the behaviors.

The term *clone agents* is used in order to describe artificial agents that can be trusted to behave according to some human being's strategy in a certain environment. That is, the closer the behavior of the clone agent to the behavior of a real person the better the clone agent. The research uses games in order to evaluate the reliability of agents. The data we are interested in is the distance, or rather, the level of closeness, between the actions taken by a person and her agent at the same decision point.

3. DESCRIPTION OF THE GAMES

In this section we detail the four games used for our experimental setting.

The Black Jack game (BJ). This game follows the rules of the classic Black Jack game. The player is dealt two cards facing up and is then offered the opportunity to take more. The dealer is dealt one faced card up and one down. The hand with the highest total wins as long as it does not exceed 21. Specifically, in our implementation, the dealer must hit until he has at least 17, regardless of what the player has. The player wins if she accumulates a hand which does not exceed 21 as long as the hand is higher than the dealer's hand or the dealer has exceeded 21. The same deck of cards is used until all the cards are dealt, then a new deck is shuffled. The player's goal is to win the game, as no stakes were involved in the game.

From the strategic point of view, this game is characterized by an opponent, though this opponent does not need to be modeled (its strategy is known). The decision space for the player is: Hit or Stand. The game involves uncertainty induced by the cards being dealt. The relative complexity of the game is derived from the fact that each game session affects the next: cards dealt can not be re-dealt (until the

[1] Excluding complex evacuation environments. A typical evacuation scenario relating to the class of problems under consideration would be getting off a plane with several exits, etc.

deck is re-shuffled) and it is difficult for the average person to calculate probabilities for the different cards based on the flow of the game. The optimal strategy for winning is counting the drawn cards and calculating the probabilities of drawing the remaining cards. The similarity level between people and the agents playing on their behalf can be measured based on the number of similar decisions, to Hit or Stand, made by both in each round of the game.

The Cost Search game (CS). In this game, the player is instructed to travel between stores in order to buy a commodity (a Television). Upon arriving at a new store, the player observes the posted price of the product. All prices are drawn from a normal distribution function with a mean and standard deviation known to the user. The player needs to decide when to stop its search and which store to go back to in order to buy the product (upon terminating the search). The search process is characterized by a finite decision horizon, i.e., the player must purchase the product before a pre-known deadline is exceeded. Visiting a new store entails a fixed amount of time (thus at each step the decision horizon can be translated into the maximum number of new stores that can be visited). In addition, the player incurs a cost for her time which is represented by the loss of income (e.g., the hourly salary that could have been earned instead of searching). The goal of the player is to minimize the overall cost of the process (the sum of the product cost and the aggregated alternative cost along the search).

From the strategic point of view, the game is played under a time constraint rather than against an opponent. Solving the game theoretically in order to extract the optimal search strategy can be done using an instance of Pandora's problem [20] resulting in a stationary threshold below which the search should be terminated. The uncertainty in the game relates to prices. The decision space for the player is simple: Continue the search or Stop and buy at the cheapest store visited. The similarity level between people and the agents playing on their behalf can be measured based on the number of similar decisions, to Continue or Stop, made by both in each game session.

The Repeated Lottery game (RL) [4, 22]. In the Repeated Lottery game the player is initially allotted a budget ($100). In each round the player needs to decide the amount of money she wishes to bet on from the current budget. Once the bet is set, the player wins with a 0.6 probability (in which case, the bet is added to her current sum) or loses with a 0.4 probability (in which case, the bet is reduced from her current sum). Each session is played for a maximum of ten betting rounds, or until the player has lost all her money. The goal of the player is to maximize the amount of money she has at the end of the session.

From the strategic point of view, the game does not include an opponent but relies on luck. The space of actions available for the player at each game step is continuous (the amount to bet) and relies on her performance in former rounds of the game (i.e., the amount of money earned up until the current point). Since the probabilities for winning and losing do not change along the game and the expected benefit of every bet is positive, the optimal betting strategy is to always bet the entire budget. Due to the continuous nature of the bet, the measure for similarity between people and their agents' actions should be based on an interval (e.g., betting the same amount up to a deviation of percentage α from the amount of money they currently have).

The Parking game (Park) [2]. In this game the player is instructed to park a car in a dynamic parking lot where cars are continually entering and exiting. The parking lot has 4 Entrance-Exit points as well as six rows of parking spaces, each with 70 aligned parking spaces, and one row aligned with 35 spaces. That is, a total of 455 available parking spaces. Traffic lanes have specific traffic direction(s). A front door located at one of the sides serves as a sole foot entrance/exit point. Foot traffic may proceed in either direction but both vehicle and foot traffic are restricted to the traffic lanes. The player is a single driver entering the parking lot, while other drivers are also looking for parking spaces. During all stages of the game, the player has limited visibility of other drivers and the available parking spaces in her vicinity. Before the game starts the player is asked to define her cost function for each parking space in terms of the importance between search time, distance from foot exit and distance from car exit. The goal of the player is to park in an available parking space while minimizing her individual cost function.

From the strategic point of view, in this game the player is affected by opponents (other drivers) operating in her environment. The complexity of the game is derived primarily from the limited visibility, but also from the existence of other drivers in the environment. There is complete uncertainty relating to the arrival and exit of cars to/from the parking lot. An optimal solution for this problem is unavailable due to both its complexity and the different cost functions representing different people with different concepts of optimality. At every time step the set of actions available to the player includes: going up/down/right/left (if direction rules allow), parking in an adjunct free space or waiting in place. The similarity between a person and her agent can be measured by comparing the actions taken at the same decision points.

4. METHODOLOGY AND EXPERIMENTAL SETTING

We programmed each of the four games described in section 3 using a client-server architecture. Clients had two forms: a GUI-based client and an agent-based client. The agent-based client was a skeleton agent[2], having all the necessary server-communication functionality (i.e., for playing the game), and lacking only the strategy layer of its implementation. All participants took part in experimenting all four games. This enabled us to analyze our results across-games and across-players. Participants were divided into two separate groups denoted "Clone" and "Optimize", and received instructions throughout the experiment based on the group to which they belong. The experiment was composed of four phases. Each phase related to all four games and was performed by all participants in parallel. The transition between phases was done only after all participants completed their task of the current phase in all four games. Instructions were given in a consecutive manner, where the instructions for each phase were given at the beginning of the phase. The remainder of this section describes the different experiment phases and gives the rationale for the experimental design used.

[2] A skeleton agent is equipped with a complete set of functionalities required for a client in the system and supplies a rich, well-defined API for adding the agent's strategy. [2]

In the first phase ("Phase I") participants were requested to thoroughly review the game rules. Then, they were instructed to "practice" playing each game, until the following two conditions were met: (a) participant feels confident in understanding the game rules, its flow and its potential outcomes; and (b) participant feels confident in having an established game strategy for the game. While all four games were based on relatively "daily" or "highly known" tasks that most participants reported to have experienced before, no exceptions were made in the intensity of the training at this phase. This was used as a precautionary measure to ensure that participants beginning "Phase II" of the experiments were fully trained and had experienced each game.

In the second phase ("Phase II") participants were asked to play at least 20 sessions of each game. All actions in the games played were logged into files. Then ("Phase III"), participants received the agent-based clients and were requested to develop their strategy layer. Participants of group "Clone" were instructed to have their agent act as closely as possible to the way they played themselves in "Phase II" of the experiment.[3] Participants of group "Optimize" were instructed to design a strategy layer that will maximize the agent's performance in the game. None of the participants at this phase had access to the log files of their "Phase II" games. Participants from group "Clone" were told they will be evaluated based on the similarity between the way their agent acts in future games and the way they acted in the games they played as part of "Phase II". Participants from group "Optimize" were told they will be evaluated based on the performance of their agents in the games. In addition to the agents themselves, the deliverables in this phase included a written explanation of the strategy used for each agent in each game.

In order for the agents to face the same decision points as their programmers, we used the log files that were produced in "Phase II". For example, the BJ "Clone" agent (and the dealer) were dealt the same cards as in each of the original rounds the agent's programmer participated. If the agent drew more cards than the person then the extra cards were randomly drawn from the remaining deck and re-inserted into the deck once the current session ended. Similarly, under-drawn cards were removed from the deck when the session ended, so that the next session began with an accurate reflection of the cards that were in the deck when the participant played her game. The agent was notified at the end of each session of the additional cards that were taken out or brought back to the deck in this session. In this manner we made sure that we recorded the actions of each agent at the exact same decision points the player faced in her games (for comparison purposes).

In a similar manner, in the CS game, the prices drawn for the product in each store when the agent played were actually taken from the log-files stored when the person who designed it played. Additional stored visits returned prices drawn from the distribution function used. In the cases of RL and Park, the situation is a bit more complex since the action taken in one decision point can affect the rest of the same session. That is, a decision to turn left instead of right or to bet $75 rather than $5 affects the comparison at the next decision points. To overcome this problem we gave the agent the opportunity to make its decision and after it was made, we forced the agent to act according to the action taken by the player at that decision point. This way, the comparison was made for each specific decision point under the same settings.

Before simulating the agents we reviewed their design and implementation. If an agent was found to be using randomization as a characteristic of its strategy (e.g., randomly choosing between two values to bet on in RL) it was executed 100 times (rather than once) for each session played by the person and the comparison metric was averaged. This was done since in case of random behavior, a single execution does not provide sufficient data regarding the closeness between the agent's and player's behaviors. The same method was used for all the agents in BJ and CS, since randomization was also used if an agent drew more cards or visited more stores than the player did.

Finally ("Phase IV"), participants from group "Clone" were asked to identify complementary tools or data they believed could have improved the performance of their agents (in terms of the observed similarity to their played strategy). Based on the responses received, participants were offered the log files collected as part of "Phase II" and to re-submit their agent. This was done in order to evaluate the effectiveness of viewing their own behavior as a means of enhancing the ability to capture one's real-life strategies. The new agents received were tested and evaluated based on the same procedure described above for the agents developed in "Phase III". The code of all the agents received in this phase were manually reviewed to prevent log-based implementation (i.e., referring to specific world states observed in the log file).

For each action taken by the player, a binary measure of similarity was used. In BJ's case, the decision was tagged identical if both the player and the agent decided to Hit or Stand at the same decision point (and otherwise it was tagged as different). If the player drew more cards than the agent, we considered the number of matching Hits and the final Stand to be identical decisions since we could be sure the agent would have chosen to stand if it was put in the same decision point where the player decided to stand. The number of unmatched Hits was considered as different decisions. Similarity in CS was measured in the same manner.

In the RL game, an action was tagged as identical if the bet made by the agent did not deviate by more than 20% from the participant's bet at each decision point.[4] For the Parking game, an action was considered to be identical only if it was taken by both the agent and the participant in a similar given setting. For all four games, the ratio between identical and overall number of actions was calculated for each session averaged over all sessions and used as the closeness measure.

At the end of the experiment, participants were requested to fill-in a detailed questionnaire that attempted to eval-

[3] While seemingly a person tries to maximize her performance in real life, there is a difference between instructing a person to design an agent that acts like her and instructing her to design an agent that maximizes payoff. This is because the computer agent can make use of capabilities that often people do not have. Therefore the instructions given to the participants in this group were to design an agent that acts like them.

[4] Due to the continuous nature of bets we had to use an interval. The 20% range was set arbitrarily. Obviously a greater percentage would have resulted in greater similarity and vice versa.

uate their initial familiarity with the different games, the programming load of developing the agents, the length of time the assignment actually took and various other aspects relating to the execution of the experiment.

Overall, 41 senior computer science undergraduates participated in our experiments, resulting in a total of 164 strategies/agents that were analyzed. The task was a part of an agent writing workshop (at Bar-Ilan University), where students were graded according to the performance measure defined for their agents. Out of the 41 participants 27 belonged to the "Clone" group and the remaining 14 were assigned to the "Optimize" group, serving as a control group. Since this research focuses on the "Clone" group, the participants were divided by a ratio of 1:2.

5. RESULTS AND ANALYSIS

Naturally, this paper's interest is in the performance of the participants belonging to the "Clone" group. We begin by introducing a brief summary of our analysis of the strategies used. Then we introduce the main results of our analysis concerning the ability of participants to clone their exhibited strategies. Finally, we report our results relating to the closeness achieved by the participants who chose to resubmit a revised agent based on observing the log files of the games they and their agents played.

Since some of the participants played more than 20 rounds in some of the games, the analysis presented takes into consideration the first 20 rounds of each game played by each participant.

5.1 Games and Strategies

Our analysis is based on the comparison of the actions of the 41 participants and the actions of their agents in similar world states (a total of 4012 rounds were played and logged during our experiments).

Based on the strategy design documents handed out by the participants as part of "Phase III" of the experiment and from manually reviewing the code handed out by each participant, we learned that all the designed strategies were general rather than case-based. Furthermore, based on the strategy design documents we extracted a list of strategy characteristics that were used in each game. While this list has many uses (e.g., for learning the complexity and richness of the strategies applicable for each game) our interest mainly involves characterizing individual strategies. Having the specific characteristics of each strategy enables us to drill down in our analysis and better understand why some people failed to accurately capture their game strategies. The main characteristics used in the different games were:

BJ: (1) threshold based, (2) dealer consideration, (3) relying on history of cards (including deck re-shuffling) and (4) randomized elements.

CS: (1) product price based, (2) passing time based, (3) predefined number of stores and (4) randomized elements.

RL: (1) predefined bets, (2) randomized elements, (3) relying on results of former betting, (4) threshold based bets and (5) never bet total current budget.

Park: (1) park in the first vacant space found, (2) park close to foot exit, (3) park in best located vacant space, (4) use randomization, (5) remember history, (6) predefined route, (7) park further away as time goes by, (8) change course if waiting too long and (9) willing to wait in place.

5.2 Closeness Performance

Table 1 presents the average closeness level of the agents programmed by participants of each group ("Optimize" and "Clone") in "Phase III" (before supplying the log files) in comparison to the games played by the participants themselves in "Phase II". From Table 1 we can see that while in some of the games the participants managed to capture their strategy in a relatively accurate measure (e.g., BJ and Park), in others the closeness measure yielded a relatively poor performance. Obviously a 100% similarity level is not expected for two main reasons. First, people's strategy always involves some "noise" (e.g., failing to remember the store with the lowest price (CS) or miscalculating the percentage of the amount of money currently at hand (RL)). Second, whenever randomization is used, in the absence of a method to observe thousands of games played by each participant, some discrepancy is most likely to occur.

The difference between the performance of the "Clone" and "Optimize" groups was tested for statistical significance by applying the statistical T-test. It proved to be significant ($p<0.001$) in RL and non-statistically significant in BJ and CS.

In Park, significance ($p<0.005$) was found in favor of the "Optimize" group. This seems confusing, since on average the "Optimize" agents managed to perform more closely to the way their programmers acted. Nevertheless, a tighter analysis of the data in this game reveals that very few agents significantly influenced the average (see below a more detailed discussion). If, for example, only participants performing above the 50% threshold are taken into consideration, then the closeness measure of the "Clone" and "Optimize" agents in the Parking game obtain a similar value.

Agent Type	BJ	CS	RL	Park
Clone	0.78	0.7	0.52	0.77
Optimize	0.75	0.69	0.41	0.82
Relative Difference	3.45%	0.6%	20.07%	-5.5%
Correlation	0.83	0.79	0.80	0.88

Table 1: Average closeness and strategy characteristics correlation between "Optimize" agents and "Clone" agents, per game.

As can be seen in the third row of Table 1 the relative difference between the closeness measure of the two groups varies. A possible explanation for this is that each game is associated with a different complexity, affecting the additional use of the agent's extended memory and computational power in its strategy. We emphasize that the game complexity discussed here is the complexity of the game for people rather than the computational complexity of the theoretical optimal strategy (the one maximizing the expected score in the game). This is nicely illustrated by the RL game. In this game, the theoretical optimal strategy is always to bet all the money the player has (due to the positive expected value of the basic lottery in each round). However, few of the participants in the "Optimize" group used this strategy (most participants actually constrained their agent not to bet all of the money in any given game phase).

Analyzing the different strategies based on the set of characteristics reported in the previous subsection, we obtain further results that complement the above findings. The last row in table 1 presents the correlation between the average use of each different strategy characteristic in the two groups. As can be seen from the results, the highest correlation was obtained in the Park game (possibly explaining the relative similarity in the closeness of agents programmed by participants from the two groups).

Overall the average closeness obtained by the "Clone" group was 69%. Nevertheless, a closer look at the specific closeness achieved in the different games reveals that a small portion of the population is responsible for most of the degradation in the average closeness. Table 2 demonstrates this phenomenon, detailing the percentage of participants that achieved a closeness score greater than the different thresholds given in its first column. From the table, we can see how the "bad strategy producers" affect the results. For example, excluding the participants (15%) that scored below a 50% closeness in Park results in a new similarity value of 0.85 (in comparison to 0.77). Generally, we can see that a relatively large portion of the population managed to achieve a closeness greater than 0.5, where the average similarity for this portion ranges between 0.74-0.85 (game-dependent).

	BJ	CS	RL	Park
closeness≥ 0.5	92%	89%	44%	85%
avg closeness	0.81	0.74	0.77	0.85
closeness≥ 0.6	92%	74%	37%	85%
avg closeness	0.81	0.77	0.81	0.85
closeness≥ 0.7	73%	52%	33%	85%
avg closeness	0.85	0.81	0.83	0.85
closeness≥ 0.8	62%	30%	22%	70%
avg closeness	0.87	0.85	0.87	0.88
closeness≥ 0.9	19%	4%	7%	26%
avg closeness	0.95	0.94	0.92	0.92

Table 2: The percentage and average closeness of participants that managed to achieve different levels of similarity.

An additional validation was performed aiming to ensure that the design of the agents programmed by the "Clone" group was not affected by a subconscious tendency to optimize performance. For this purpose, we compared the game performance (i.e., the actual score achieved) in the games played by the agents versus the games played by the people who programmed them (in the "Clone" group). The analysis did not reveal any significant dominance of the agents' performance (game score) over the participants' performance.

As depicted in Table 2, there is a segment of the population (per each game) that cannot translate its strategy patterns into structured strategies which can be embedded in agents for various reasons. Consequently, the rest of the analysis focuses on understanding what characterizes people from the latter segment and how they can be identified.

Since the average closeness is different in the different games we used ranking to normalize the data. For each game all agents were sorted according to their closeness value and assigned an integer number between 1 and 27 (the agent with the highest closeness value was ranked 1).

While different participants were ranked in different positions in different games, we were interested in finding evidence of the existence of people who are generally better than others in capturing their strategies. Table 3 shows for each game the average ranking in the other three games of its five top ranked participants. For example, the top 5 ranked participants in BJ were averagely ranked 15 based on their closeness measures in the other three games. The table also provides the range (i.e., min,max) of the closeness ranking obtained by the top 5 ranked participants in each game. For example, the rankings in the other three games of the 5 top ranked participants in BJ varied between 5 and 26. From the table we learn that there is no group of people who are generally better than others in capturing their own strategies. The people who did best in one game did not replicate this achievement in the other games, and in many cases were ranked in the lower portion of the scale.

	BJ	CS	RL	Park
Average ranking in other games	15	16	12	12
Range of Ranking	5-26	2-27	2-27	1-24

Table 3: Ranking in other games of the top 5 participants in each game

Lastly, based on the analysis of the questionnaires, we attempted to understand how a participant's reported estimations of the closeness of her agent and her strategy completeness would predict her performance in producing clone agents. The first row in Table 4 presents the correlation coefficient between the participants' estimations of the results of their agents and the actual results obtained (per game). The second row in this table, details the average ranking of people who reported their strategy as complete and mature when they started playing in "Phase II" versus the average of those who reported their strategy as not fully established.

	BJ	CS	RL	Park
Performance prediction	27%	-9.5%	5.8%	6%
Strategy completeness (partial / full)	11 / 14.5	16.6 / 12.85	14.4 / 15.7	9.3 / 14.43

Table 4: A person's reported closeness prediction and level of understanding of the game as a predictor of closeness.

As portrayed in the table, people's prediction of the closeness performance of their agent cannot be used as a general predictor. As for a person's strategy completeness, from an analysis of the questionnaires we found that on average (per game) 10 people reported some level of incompleteness in their strategy. This was despite our thorough preparation phase and practice games that were defined in the experimental design. Nevertheless, as evident from Table 4, there is no consistency or significant difference in the ranking of the two groups. Even when a person tends to believe her strategy is not fully established, her ability to reproduce her strategy is of the same level as that of people who believe their strategy is fully established.

Overall, the results strengthen our findings that people can reasonably capture their strategies in the class of games we investigated.

5.3 Enhancing Closeness by Supplying Information

As described in the previous section, in an attempt to improve participants' ability to better capture their game strategy, we allowed them (in ”Phase IV”) to make use of the log files of their games (played by themselves and by their agents) in order to review their strategy. This stage was optional, and of 27 participants, 7 decided to fine tune the strategy layer of their BJ agent, 13 in CS, 16 in RL and 7 in Park. The first two rows in Table 5 present a comparison between the average closeness of these participants (per game) in their two agent versions. As depicted in the table, by viewing the discrepancies between their own and their agent's actions, participants did manage to improve the closeness between their behavior and their agents' behavior in all 4 games.[5]

	BJ	CS	RL	Park
Closeness Phase III (improvers)	0.74	0.658	0.5	0.76
Closeness Phase IV (improvers)	0.76	0.659	0.58	0.83
Avg rank of impr.	14.4	16.7	14.2	17.43
Avg rank of non-impr.	13.2	11.5	13.7	12.8

Table 5: Average closeness and ranking of the participants who chose to try and improve their agents.

Testing for significance between the improvers' performance in phases III and IV while using the paired t-test shows significance for both RL (p<0.001) and Park (p<0.001). However, the significance between the ”Clone” and ”Optimize” groups' performances in Park remains significant (p<0.001) in favor of the ”Optimize” group, even when the improved cloned agents are taken into consideration.

It is worth noting that despite the fact that participants were not aware of their relative ranking in the experiment (i.e., the level of similarity their agents exhibited in comparison to other agents) those who had lower ranks decided to improve their agents based on the new information disclosed. While this might seem obvious, one must keep in mind that given the results presented in the last two rows of Table 5 all agents had room for improvement.

Another interesting finding relates to participants who used randomization in their strategies: 70% of the participants who resubmitted their agents used randomization in their strategy. Thus, in comparison to their relatively small portion in the general population (28% in ”Phase III”), this might suggest that the method is beneficial in particular for people who use randomization in their strategies.

5.4 Building the Agents

Following the completion of the experiment each participant was asked to answer a questionnaire. Due to space limitation, we report the answers to two questions which are of special interest in this context. The first question concerns the tendency of participants to play differently in phase II if they would have known the assignment in phase III in advance. Only 11% of participants reported that they would have played differently in this case. This relatively low percentage may suggest that people are confident with their strategies and in their ability to capture them. The second question concerns the participants' opinion regarding the ability of people to capture their strategies and write clone agents. Here, again, only 11% of participants responded negatively.

6. RELATED WORK

As discussed in the introduction, the need to capture individual behaviors often arises in the design of agent-based environments (e.g., simulation systems [21]). Here, we can find various implementation methods for modeling individuals' behaviors, e.g., using statistical data [17], pre-specifying agents' roles using a set of parameters according to which the agents act [11], or using a combination of rules and finite state machines for controlling an agent's behavior using a layered approach [19]. The main difference between these applications and our approach is that they all leave the task of simulating the individual behaviors to the simulation designer or domain experts. We, on the other hand, attempt to extract behaviors directly (and reliably) from a representative portion of the population. Recent research suggests methods by which an agent can learn behaviors from observing peoples' games [16]. However, these methods require substantial data which the agent does not often have.

Several works have used people to design agents that can represent them in a MAS and act on their behalf. For example, Kasba [3] is a virtual marketplace on the Web where people create autonomous agents in order to buy and sell goods on their behalf. Another example is the use of people for programming agents under the decision-theoretic framework of the Colored-Trails game [8]. Here, the agents had to reason about other agents' personalities in environments in which agents are uncertain about each other's resources. Nevertheless, in these implementations no attempt has been made to test how reliably these strategies represent the people who programmed them.

Evidence of discrepancies between actual and reported behavior is a prevalent theme in research originating in other various domains. In particular, metacognition research[6] commonly emphasizes the limited extent to which people are aware of the strategic ”policies” which guide their decision-making behavior. For example, doctors were found to believe their treatment decisions are affected by various factors that they do not, in fact, take into account when making their decisions [9]. As a result, they state that they need information which they do not actually use.

Other examples include: (a) Over-reporting of political participation (roughly 25% of non-voters report of having voted immediately after an election [1]) and (b) Contrasting results between self-reported and performance-based levels of physical limitations (there is weak to moderate association between performance-based and self-reported measures in motor functioning [10]). Many possible explanations have been suggested for this phenomenon. For example, some researchers believe that on top of the cognitive problem, social desirability affects peoples' answers as well [1], i.e., respondents want to avoid looking bad in front of the interviewer. Others, see socio-demographic variables, cognitive functioning, affective functioning and personality characteristics as possible factors [10].

While the above works provide evidence of differences between reported behavior and actual behavior, they differ from our work in reference to two aspects. First, our work initially asserts that there are some inherent differences be-

[5] As described in section 5.1, all agents' strategy layers were manually checked in order to make sure none of the enhancements related to specific cases.

[6] Metacognition is thought about thought, often characterized as ”self insight” [18].

tween the two and focus on the attempt to measure the level of difference. Second, the behaviors reported in earlier research are generally simpler and do not apply to strategic decision making of the type we investigate.

7. DISCUSSION, CONCLUSIONS AND FUTURE RESEARCH

As discussed in the introduction and in the related work sections, there is a great challenge in developing a set of behaviors with enough variety and realism that can reliably represent the richness of behaviors observed in real-life environments. The process of distributing the task of strategy elicitation to the people whose strategies we are trying to capture and embed in agents can significantly change the way MAS are designed and built, in terms of cost, speed and reliability.

This paper relates to a class of environments represented by four games. The selected games share many characteristics, yet each game emphasizes specific ones as discussed in Section 3. Generalization to more complex games requires further experimentation using the methodology presented in this paper. Nonetheless, even in extremely complex environments the use of traditional learning mechanisms can benefit from applying the proposed methodology as a preliminary step.

The analysis presented in Section 5, suggests that generally, a large portion of the population is capable of capturing their exhibited strategy for the class of environments under question with a reasonable precision level. This is not sufficient for generally declaring a simulation built on self-programmed agents as a reliable reflection of the "real-life" environment. However, it is good evidence for the strength of this approach if managed properly. Furthermore large-scale experimentation is required in order to produce insights that can help identify what can and cannot be used to predict the success of a person in capturing her strategy. In the current research we show evidence that a person's own evaluation of her strategy's completeness and similarity performance cannot be used as good predictor of her success in capturing her own strategy. The results do show evidence of the usefulness of an iterative process by which participants observe the discrepancies between their exhibited and declared actions as a means of improving the reliability of the strategies produced.

These are the first steps in the long journey to understanding how people perceive their strategic behavior and what means should be applied in order to help them elicit their strategies. To the best of our knowledge, this is the only large-scale attempt of this type made so far to measure the ability of people to elicit their strategic behavior. Future research should focus on testing additional/alternative methods for enhancing participants' ability to reliably capture their strategies.

8. REFERENCES

[1] M. Bertrand and S. Mullainathan. Do people mean what they say? implications for subjective survey data. *American Economic Review*, 91(2):67–72, 2001.

[2] M. Chalamish, D. Sarne, and S. Kraus. Mass programmed agents for simulating human strategies in large scale systems. In *AAMAS*, 2007.

[3] A. Chavez and P. Maes. Kasbah: An agent marketplace for buying and selling goods. In *PAAM*, 1996.

[4] J. Eichberger, W. Güth, and W. Müller. Attitudes towards risk: An experiment. *Metroeconomica*, 54(1):89–124, February 2003.

[5] H. C. T. Force. A report on the use of traffic simulation models in the san diego region. Technical report, Institute of Transportation Engineers, Highway Capacity Task Force, 2004.

[6] H. Futura and M. Yashui. Evacuation simulation in underground mall by artificial life technology. In *ISUMA*, 2003.

[7] N. Griffiths and M. Luck. Coalition formation through motivation and trust. In *AAMAS*, 2003.

[8] B. Grosz, S. Kraus, S. Talman, B. Stossel, and M. Havlin. The influence of social dependencies on decision-making: Initial investigations with a new game. In *AAMAS*, 2004.

[9] C. Harries, J. S. Evans, and I. Dennis. Measuring doctors' self-insight into their treatment decisions. *Applied Cognitive Psychology*, 14:455–477, 2000.

[10] G. Kempen, M. van Heuvelen, R. van den Brink, A. Kooijman, M. Klein, P. Houx, and J. Ormel. Factors affecting contrasting results between self-reported and performance-based levels of physical limitations. *Age and Ageing*, 25(6):458–464, 1996.

[11] D. Massaguer, V. Balasubramanian, S. Mehrotra, and N. Venkatasubramanian. Multi-agent simulation of disaster response. In *ATDM AAMAS*, 2006.

[12] D. Massaguer, V. Balasubramanian, S. Mehrotra, and N. Venkatasubramanian. Synthetic humans in emergency response drills. In *AAMAS*, 2006.

[13] Y. Murakami, K. Minami, T. Kawasoe, and T. Ishida. Multi-agent simulation for crisis management. *IEEE International Workshop on KMN*, 2002.

[14] Y. Murakami, Y. Sugimoto, and T. Ishida. Modeling human behavior for virtual training systems. In *AAAI*, 2005.

[15] M. Rabin. Psychology and economics. *Journal of Economic Literature*, 36(1):11–46, March 1998.

[16] G. Sukthankar and K. Sycara. Policy recognition for multi-player tactical scenarios. In *AAMAS*, 2007.

[17] T. Takahashi, S. Tadokoro, M. Ohta, and N. Ito. Agent based approach in disaster rescue simulation - from test-bed of multiagent system to practical application. In *RoboCup*, 2002.

[18] M. Twyman, C. Harries, and N. Harvey. Learning to use and assess advice about risk. *Forum: Qualitative Social Research*, 7(1), 2006.

[19] B. Ulicny and D. Thalmann. Towards interactive real-time crowd behavior simulation. *Computer Graphics Forum*, 2002.

[20] M. L. Weitzman. Optimal search for the best alternative. *Econometrica*, 47(3):641–54, May 1979.

[21] Y. Zhang, K. Biggers, L. He, S. Reddy, D. Sepulvado, J. Yen, and T. Ioerger. A distributed intelligent agent architecture for simulating aggregate-level behavior and interactions on the battlefield. In *SCI*, pages 58–63, 2001.

[22] B. J. Zikmund-Fisher. De-escalation after repeated negative feedback: emergent expectations of failure. *JBDM*, 54:365–379, 2004.