Neurocomputing 114 (2013) 92–97

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Design of social agents

Roman Gorbunov*, Emilia Barakova, Matthias Rauterberg

Eindhoven University of Technology, Den Dolech 2, Postbus 513, 5600 MB, Eindhoven, The Netherlands

ARTICLE INFO

Available online 5 November 2012

Keywords:
Social interactions
Agent based modeling
Game theory
Evolutionary game theory

ABSTRACT

Social behavior, as compared to the egoistic behavior, is known to be more beneficial to groups of subjects and even to individual members of a group. For this reason, social norms naturally emerge as a product of evolution in human and animal populations. The benefit of the social behavior makes it also an interesting subject in the field of artificial agents. Social interactions implemented in computer agents can improve their personal and group performance. In this study we formulate design principles of social agents and use them to create social computer agents. To construct social agents we take two approaches. First, we construct a social computer agent based on our understanding of social behavior. Second, we use an evolutionary approach to create a social agent. The social agents are shown to outperform agents that do not utilize social behavior.

1. Introduction

Collaboration, trust and other social norms are crucial for good operation of teams. Our long term goal is to create a model of social norms that will enable prediction of behavior and interactions within a team. On the basis of such a model we want to test hypotheses of how these interactions within a group will develop.

Human and animal models of behavior inspired studies on social learning [4,18–20,22] and models of social norms such as trust and collaboration [18–20,15]. Collaboration has also been studied within economics, sociology, and game theory. In economics, models of a rational actor who collaborates to improve organizational effectiveness have been developed [16,12,10]. In sociology [5], the non-rational and social aspects of collaboration has been explored.

Games, in particular are an establishing field that makes possible realistic social interactions to be exposed, monitored and trained [1–3,13,14,17]. Hennes et al. [13], Voynarovskaya et al. [21] and Gorbunov et al. [8] have developed a game-based method for monitoring the social relations in small groups of individuals. It is based on the records of a negotiation game that monitors of the collaborative behavior of the players. The Colored Trails game framework [7] in a three-player negotiation variation [6,21] was used for this study. The conclusions of the study have been restricted by the limited amount of data and the difficulties to constrain the experiment by having a control over one or several individuals.

We propose to use a multi-agent system approach for the purpose of creating an agent-based support tool for team collaboration as a first step towards prediction of conflicts in small teams of people. We design the agent-players that in controlled manner can express collaborative or egoistic behavior. We hypothesize that the agent with persistent collaborative behavior will be more successful in collaboration game. We also expect that the outcome of the experiments will give us better insights of how to improve the agents so they can be used as a controlled substitute of a human player.

Moreover, we want to understand the mechanism of collaborative behavior, i.e. how collaboration is maintained, how to avoid exploitation of collaborative agents. This understanding will help us interpret the data from MARS 500 experiment [13,21,11].

2. Methods

2.1. Settings of the interactions

In our study we use the prisoner dilemma, a classical problem in the game theory, which used to study collaborative behavior. Every player can chose either cooperation or defection. If both players choose to cooperate, than they get three points each. In contrast, every player gets only one point if both of them choose to defect. In other words, the mutual cooperation is more beneficial than the mutual defection. On the other hand, the defection is always more beneficial than then cooperation under any given strategy of the partner. If one player defects and another one cooperates than the defecting and cooperating player get 5 and 0 points, respectively. Because of the above mentioned

* Corresponding author.
E-mail address: r.d.gorbunov@gmail.com (R. Gorbunov).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.06.046

properties of the game, players face a dilemma: to cooperate or to defect.

In our study the interaction between two players (computer agents) is organized in the following way. One of the agents starts the game by choosing one of the two available moves (cooperation or defection). Another agent "sees" the move chosen by the first agent and uses this information to make its own move. After two agents made their moves, each of them gets points determined by the above given payoff matrix. The third move is made by the first agent. The third and the second moves together determine another portion of points that will be delivered to the two interacting agents. In other words, the second move determines the payoffs two times (first, in combination with the first move and then in combination with the third one). We use the repeated setting of the prisoners dilemma in which the probability of next interaction to happen is fixed. The value of this probability has been set to 0.995 to make the average number of games in a sequence of interactions to be equal to 200.

2.2. Model of the agents

To formalize algorithms imbedded into the computer agents we consider a model similar to the Turing machine. Every agent has a fixed and limited number of internal states. The previous move of the partner is considered as the input for the algorithm. The input and the internal state of the agent determine its output (the next move) that can be either cooperation or defection. The mapping of the inputs and internal states into the output is given by a fixed matrix. Additionally to the generation of the output the agent changes its internal states. The new internal state, like the output, is determined by the input and previous internal state. The mapping of the input and the current internal states into the new internal states is also given by a fixed matrix.

2.3. Design principles

The maximization of the personal benefit of an agent in a sequence of interactions with a random partner is the main design criterion that we used for the construction of the social computer agents. In other words, our intention is to create an egoistic agent whose the only goal is to maximize its own benefit in the environment populated by other agents whose playing strategies are not known.

In the prisoner's dilemma the defection is always the most beneficial choice independently on what choice, defection or cooperation, is made by the partner. As a consequence, the agents, that are rational in the classical game theory sense, always choose to defect. In other words, the rational agents demonstrate mutual defection that is known as Nash equilibrium for the prisoner's dilemma. Further on we will call the agents of this kind as "defective" agent.

The permanent defection is one of the two simplest possible strategies and the permanent cooperation is the second one. We will call the agents that permanently cooperate as "cooperative" agents. This strategy can also be considered as a rational one since the mutual cooperation is more beneficial than the mutual defection.

However, it has to be emphasized that the cooperation is beneficial only if it is reciprocal. The cooperation with an agent that defects can be considered as an exploitation of the cooperative agent by the defective one. This kind of interaction is very beneficial to the exploiting agent and very unbeneficial to the exploited one. Thus, the cooperative agent cannot be considered as a successful solution for the environment populated by agents of different kinds.

To prevent an exploitation by a partner the agent should exhibit a reciprocal behavior (defect with the defective agents and cooperate with the cooperative agent). We will call the agents of this type as "reciprocal" agents. The reciprocal agents can be considered as more successful than the above introduced defective and cooperative agents. In contrast to the cooperative agents, the reciprocal agents cannot be exploited by the defective agents since they will behave defectively with the defective agents. In contrast to the defective agents, the reciprocal agents will establish and maintain cooperation with each other if they are constructed to start from the cooperative move. The weak point of the reciprocal agents is that they are not able to exploit cooperative agents. In other words, the reciprocal agent cooperates with its partner even if it is not a necessary condition for the cooperation of the partner. This kind of behavior is in a disagreement with the above introduced goal to find a behavior that maximizes the personal benefit of the agent.

The above described requirements can be summarized in the following list of the social skills that have to be embedded into the social agents:

- To avoid the exploitation by the partner, the social agent should be able to reciprocate defection of the partner (react defectively on the defection).
- To maintain the cooperation, the social agents should be able to reciprocate the cooperation of the partner (react cooperatively on the cooperation).
- To establish the cooperation, the social agents should be able to propose the cooperation in the phase of the mutual defection (react cooperatively on the defection).
- To exploit the partner, the social agents should be able to try defection in the phase of the mutual cooperation (react defectively on the cooperation).

As we can see, the above described requirements can be competing. So, the agent needs to be able to decide what logic to apply at the current stage of the interaction. Moreover, the agent needs to remember the responses of the partner. In more details, it should not keep trying to establish a cooperation if all these attempts fail. Similarly, the agent should not continuously try to exploit its partner if it sees that these attempts are not successful. On the other hand, the agent should not cooperate if it is not required for the cooperation of the partner.

We implement the above described design principles by the following algorithm. The agents start cooperation by proposing cooperative move as a response on the defective move of the partner. Doing that agent switches to the state of cooperation (*CN* in Fig. 1). If partner answers defectively on cooperation initiated by the agent, the agents switches to the cooperation-failed (CF_n) states. In this state the agents never initiates a new cooperation. If partner starts cooperation, the agent will defect, immediately or after several steps of cooperation, to check if its partner can be exploited. In more detail, for several steps agent plays only defectively. If at this stage the partner made at least one cooperative move, the agents concludes that exploitation was successful and try to exploit the partner again. If the agent has already tried to exploit after n steps of cooperation and this attempt failed, the next attempt to exploit will start after $n+1$ steps of cooperation. If the agent tried to exploit after $N_{\max}$ steps of mutual cooperation and this attempt failed, it never tries to exploit again. After every unsuccessful attempt of exploitation agents tries to establish cooperation. More details about the algorithm are given in Table 1. In this table different columns and rows represent different inputs (moves of the partner) and internal states of the agents, respectively. The cells of the tables contain outputs (moves of the agents) and a new state of the agent. The defective and cooperative moves are denoted as $\ominus$ and $\oplus$, respectively.

The presented constructed agent is an improved version of the agent presented in our earlier work [9]. The new constructed

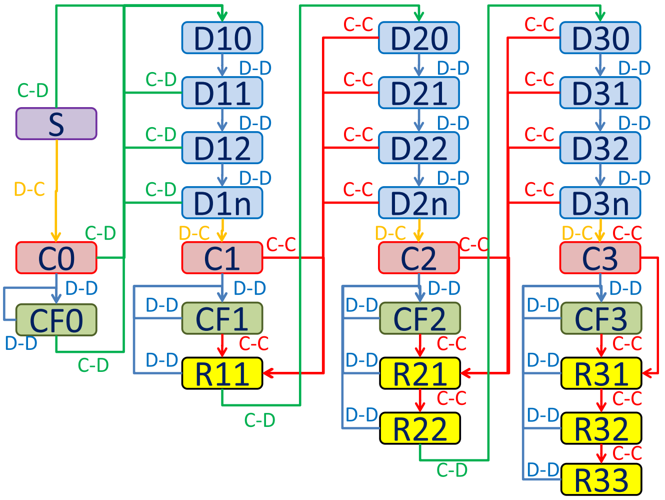

Fig. 1. Schema of the constructed social agent.

Table 1
Algorithm of the constructed social agent.

Internal state	Cooperative input	Defective input	No input
S	⊖, D10	⊕, C0	⊕, C0
D10	⊖, D10	⊖, D11	
C0	⊖, D10	⊖, CF0	
D11	⊖, D10	⊕, C1	
CF0	⊖, D10	⊖, CF0	
C1	⊕, R11	⊖, CF1	
CF1	⊕, R11	⊖, CF1	
R11	⊖, D20	⊖, CF1	
D20	⊕, R11	⊖, D21	
D21	⊖, D20	⊕, C2	
C2	⊕, C2	⊖, CF2	
CF2	⊕, CF2	⊖, CF2	

agent, presented in this work, tries to exploit its opponent even if the agents are in a stage of mutual cooperation which was initiated by the constructed agent.

2.4. *Evolution of agents*

In our study we apply an evolutionary optimization to find a well performing agent that can be compared with the constructed one. Agents with different number of states are evolved independently. We consider agents which have from 1 to 12 internal states. The evolution starts from a random agent which we refer to as a “winner”. At every step of the evolution we compare the performance of the “winner” with three other agents. One of them is just a random agent which has the same number of internal states as the “winner”. Additionally to that we consider an agent which is obtained by a small mutation of the “winner”. During the mutation we change one element in one of the two matrices that define the behavior of the agent. And finally we consider a doubly mutated agent which is obtained by a mutation of a singly mutated agent. The performances of the four agents in a pool consisting of 10 thousands random gents are calculated and compared. The agent with the highest performance is considered as the ‘‘winner’’ on the next step of the evolutionary process.

2.5. *Performance of agents*

Payoff of two agents interacting with each other depends on number of interactions in the sequence of interactions and this number is not fixed. The only thing that we fixe is the probability that there will be one more interaction. The sequence of the interactions can be terminated after any interaction. In our study, when two agents interact with each other we average their payoffs over all possible lengths of interactions taking into account the probabilities that the given length will be observed.

The performance of the agents also depends on which of the agents made the first move. To get a symmetric treatment of any two agents we let each of them to start the sequence of interaction and then average the payoff of two sequences.

The performance of an agent also depends on the partner. Moreover, even if agent *A* outperforms agent *B* while both playing with partner *C* it is still possible and agent *A* underperforms *B* while both playing with another partner *D*. To get a representative performance of an agent we create a large pool of random partners and let the agent to play with every partner from the pool. Then the performances of the agent with different partners are averaged to get the representative performance. In our study we populated the pool by agents that have not more than 12 internal states.

The size of the pool has to be larger enough to ensure that performance of the agent is the same for different pools of the same size. In our calculations we used a pool of 10 thousands agents. To estimate the accuracy associated with this size of the pool we have considered performance of a random agent in 10 different pools of this size. Because pools are different the same agent performs differently depending on the pool. However, if the size of the pool is big enough this difference is expected to be small. To measure this difference we have compared the range of the performances (difference between the maximal and minimal performance) with the average performance of the agent. We found out that the range of the performances was less the 2% of the average performance.

3. Results

A pool consisting of 10 thousands agents has been used to estimate the performance of the constructed agent. We found out that the score per game for this agent is equal to approximately 635 points. To get an idea of how good this performance is we have calculated performances of 10 thousand randomly generated agents. The distribution of the performances of these agents is shown in Fig. 2. As we can see in the figure, the constructed social agent outperform majority of randomly generated agents. In more detail, approximately 98.7% of random agents were outperformed by the constructed agent.

In case of the evolved agents the performance has a tendency to grow as the number of the internal states grows. In Fig. 3 we can see the performance of the evolved agents, obtained after 740 steps of evolutions, as a function of the number of their internal states. As we can see in the figure, in the pool of random agents the evolved agents perform better than the constructed one if the number of internal states is larger than 4. However, as we will show later, there are other ways to estimate performances of the agents and for them the constructed agent over-performs the evolved ones.

We have to note that the performances calculated in the above described way should depend on the way the pool is populated. In our study we used a procedure in which any two agents, that have the same number of the internal states, have the same chance to be put into the pool. Moreover, the probability of an agent to be added to the pool does not depend on the number of the internal states. In other words we got a “uniform” distribution of the agents in the pool and, as a consequence, the performance of agents, calculated in this way, can be considered as more “representative” or “universal”.

However, we need to remember that the assumption of the homogeneous distribution of the agents in the pool is not always valid. In particular, the pool can be populated by artificial agents that were constructed with the intention to increase the personal performance. Or, as another example, the content of the pool can be a product of an evolution. In both cases the chances of an agent to be in the pool are higher if its performance is better.

To overcome the problem of the dependency of the score on the way the pool of partners was generated, we propose to compare performances of two agents by letting them to interact with each other. In more detail, we want to model performances of two different agents in the pool that is equally populated by the agents of the two considered types. To model this situation we do not need to use a larger pool because the interactions of a given agent with the partners of the same type will always give the same score per game. The agent of a chosen type needs to interact with one partner of the same type and one partner of another type. Interactions with partner of two different types of agents gives two values of the performance that are then averaged to get the performance of the given agent in the considered situation. The performances of the two agents, calculated in this way, can be interpreted as their abilities to survive, in the evolutional sense, in the pool that is equally populated by the agents of the two considered types.

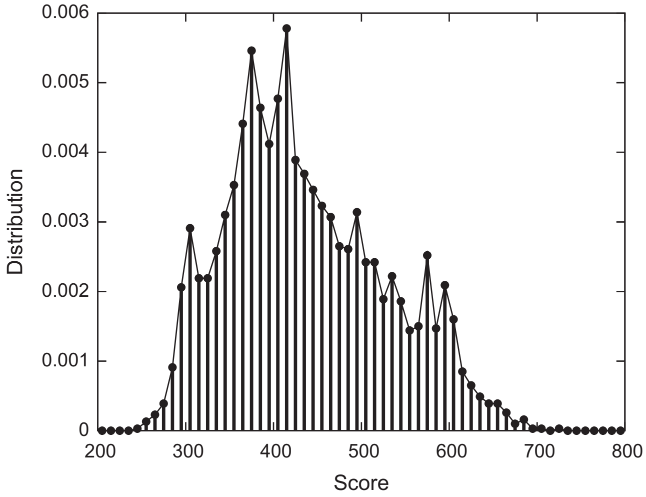

Fig. 2. Distribution of the performances of the random agents.

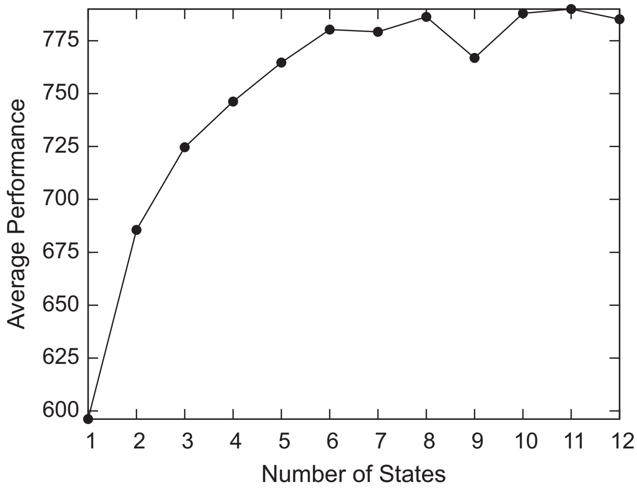

Fig. 3. Performance of the evolved agents, obtained after 740 steps of evolutions, as a function of the number of their internal states.

We would like to emphasize that the performances of agents, calculated in the above described way, are not transitive. If agent *A* outperforms agent *B* and agent *B*, in its turn, outperforms agent *C*, it does not necessarily means that agent *A* outperforms agent *C*. As the consequence, the second way to compare performances of agents cannot be used in the evolutionary optimization.

We used the above described procedure to estimate performances of the constructed and evolved agents. The constructed social agent has been compared with 1000 randomly generated agents. In more detail, for every pair of the random and constructed agent we calculated the performances of the both agents while interacting with each other. The obtained performances are shown in Fig. 4. The *x*-axis and *y*-axis correspond to the scores of the constructed and random agents, respectively. As we can see in the figure the performance of the constructed agent was always better than those of the confronted random partner. This represents the fact that the constructed agents cannot be exploited by their partners.

In the same way we have estimated the performance of the best agents found by the evolutionary optimization (see Fig. 4). As we can see in the figure, the random agents outperformed the evolved one in many cases. In other words the evolved agent does not have mechanisms that protect it from the exploitation.

Another way to compare two given agents is to study how their performances depend on their proportions in the population consisting only of these two agents. Depending on these dependencies four qualitatively different cases are possible.

- Independently on the proportions of the agents, one of the agents has a higher average payoff than the other one. As a consequence, the weakest agent will become extinct as a result of the evolution independently on the initial proportions of the two agents in the population. In this case we can say that one agent dominates another one.
- It can be the case that the agents have larger payoffs if their proportions in the populations are small enough. In this case the proportion of the agents with higher payoff increases until

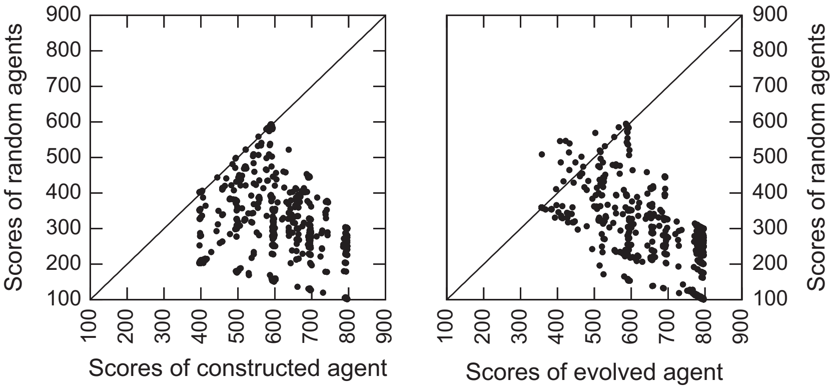

Fig. 4. Performances of the constructed and evolved agents compared with the performances of the random agents.

it reaches the point at which both agents have the same payoff. This point is called convergence point.

- We also can have a situation which is opposite to the previous one. An agent has a higher payoff only if its portion in the population is larger enough. In this case the agent whose initial portion in the population is not large enough will become extinct as a result of the evolution. We will call this case as a case of divergences.
- And finally we have a trivial case in which payoffs of two agents are always equal to each other independently on the proportions of these agents in the population.

We have considered qualitative relations the best evolved agents with 10,000 random agents. The best evolved agent dominated 8741 random agents. In eight cases the evolved agent had a convergence point with a random agent. It means there are random agents that can invade a population consisting of evolved agents but the evolved agents will not become extinct. The agents of two types will coexist. In 1250 cases the random agents had a case of the divergence with the evolved agent. In these cases the random agents can suppress the evolved agents only if their portion in the populations is larger enough.

In case of the constructed agent we did not see any cases of the attraction and absolute domination by a random agent. In more detail, in 7593 cases out of 10,000 the constructed agent dominated a random one and in 2407 cases the evolved and random agents had a case of divergence. This means that the constructed agents, in contrast to the evolved one, cannot be invaded by a random agent.

4. Discussion

In our study we provided computer agents that are able to increase personal performance by utilizing social interactions with random agents with unknown strategies. In more details, we provided a mechanism that allows agents to establish and maintain cooperation. The presented agents establish and maintain cooperation only if it is necessary and sufficient condition for the cooperative behavior of the partner. Moreover, the constructed agents are able to prevent exploitation by the defective partner. And finally, the presented agents can exploit those partners that allow the exploitation. We have demonstrated that the constructed agents perform outperform majority of random agents and even the agents that have been obtained by evolutionary optimization. In this way we proposed the computational model that describes cooperation and exploitation.

The proposed model provides a mechanism of how collaboration and exploitation can be established and maintained. As a consequence we contribute to a better understanding of these social phenomena. The presented model can be used to identify collaborative and exploiting patterns in humans behavior and, in this way, explain human behavior better. Moreover, the constructed agents mimic human social behavior and, as a consequence, can better play against human players in collaborative computer games.

Acknowledgments

The research reported in this paper is supported by NWO User Support Program Space Research. The project number is ALW-GO-MG/07-13.

References

[1] E.I. Barakova, J. Gillessen, L. Feijs, Social training of autistic children with interactive intelligent agents, J. Integr. Neurosci. 8 (1) (2009) 23–34.
[2] E.I. Barakova, T. Lourens, Expressing and interpreting emotional movements in social games with robots, Pers. Ubiquit. Comput. 14 (2010) 457–467.
[3] T. Bekker, J. Sturm, E. Barakova, Design for social interaction through physical play in diverse contexts of use, Pers. Ubiquit. Comput. 14 (5) (2010) 381–383.
[4] K.E. Bonnie, V. Horner, A. Whiten, F.B.M. de Waal, Spread of arbitrary conventions among chimpanzees: a controlled experiment, Proc. Biol. Sci. 274 (1608) (2007) 367–372.
[5] P.J. DiMaggio, W.W. Powell, The iron cage revisited: institutional isomorphism and collective rationality in organizational fields, Am. Sociol. Rev. 48 (1983) 147–160.
[6] S.G. Ficici, A. Pfeffer, Modeling how humans reason about others with partial information, in: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, vol. 1, 2008, pp. 315–322.
[7] Y. Gal, B.J. Grosz, S. Kraus, A. Pfeffer, S. Shieber, Colored trails: a formalism for investigating decision-making in strategic environments, in: Proceedings of the 2005 IJCAI Workshop on Reasoning, Representation, and Learning in Computer Games, 2005, pp. 25–30.
[8] R. Gorbunov, E. Barakova, R. Ahn, G.W.M. Rauterberg, Monitoring Interpersonal Relations Through Collaborative Computer Games, submitted for publication.
[9] R. Gorbunov, E. Barakova, M. Rauterberg, Design of social agents, Lect. Notes Comput. Sci. 6686 (2011) 192–201.
[10] R.M. Grant, The resource-based theory of competitive advantage: implications for strategy formulation, Calif. Manage. Rev. 34 (1991) 119–135.
[11] V. Gushin, Psychological countermeasures during space missions: Russian experience, J. Gravit. Physiol. 9 (1) (2002) 311–312.
[12] G. Hamel, Y.L. Doz, C. Prahalad, Collaborate with your competitors, and win, Harv. Bus. Rev. (1989) 133–139.
[13] D. Hennes, K.P. Tuyls, M.A. Neerincx, G.W.M. Rauterberg, Micro-scale social network analysis for ultra-long space flights, in: The IJCAI-09 Workshop on Artificial Intelligence in Space, Pasadena, California, USA, 2009.
[14] D. Hennes, K.P. Tuyls, G.W.M. Rauterberg, State-coupled replicator dynamics, in: 8th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2009), 2009, pp. 789–796.
[15] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, P. Li, K.W. Lee, Bio-inspired multi-agent collaboration for urban monitoring applications, Lect. Notes Comput. Sci. 5151 (2008) 204–216.
[16] J. Pfeffer, G.R. Salanick, The External Control of Organizations, Harper and Row, New York, 1978.
[17] M. Rauterberg, M. Neerincx, K. Tuyls, J. van Loon, Entertainment computing in the orbit, Int. Fed. Inf. Process. 279 (2008) 59–70.
[18] M. Tomasello, The Cultural Origins of Human Cognition, Harvard University Press, Harvard, 1999.
[19] D. Vanderelst, R.M. Ahn, E.I. Barakova, Simulated trust: towards robust social learning, in: Artificial Life XI: Proceedings of the 11th International Conference on the Simulation and Synthesis of Living Systems, 2008, pp. 632–639.
[20] D. Vanderelst, R.M. Ahn, E.I. Barakova, Simulated trust: a cheap social learning strategy, Theor. Popul. Biol. 76 (2009) 189–196.
[21] N. Voynarovskaya, R. Gorbunov, E. Barakova, R. Ahn, M. Rauterberg, Nonverbal behavior observation: collaborative gaming method for prediction of conflicts during long-term missions, in: ICEC 2010, Lecture Notes in Computer Science, 9th International Conference on Entertainment Computing, vol. 6243, 2010, pp. 103–114.
[22] A. Whiten, A. Spiteri, V. Horner, K.E. Bonnie, S.P. Lambeth, S.J. Schapiro, F.B. de Waal, Transmission of multiple traditions within and between chimpanzee groups, Curr. Biol. 17 (12) (2007) 1038–1043.

Roman Gorbunov received the M.S. in theoretical physics (2000, cum laude) at the Dnipropetrovsk National University (Ukraine). He also has Ph.D. in quantum chemistry (2007) from Goethe University (Frankfurt, Germany). From 2007 to 2009 he worked as postdoctoral research associate at Biophysics and Physiology Department at Albert Einstein College of Medicine, Yeshiva University (New York, USA). From 2010 to 2012 he worked on his second doctoral theses in the field of behavioral game theory and data mining on affective data at Designed Intelligence Group, Department of Industrial Design, Eindhoven University of Technology (The Netherlands).

Emilia Barakova is affiliated with the Department of Industrial Design at the Eindhoven University of Technology, The Netherlands, and simultaneously holds a visiting researcher position at RIKEN Brain Science Institute in Japan. She has masters degree in electronics and automation from Technical University of Sofia (Bulgaria) and Ph.D. in mathematics and physics from Groningen University (The Netherlands, 1999). She has worked at RIKEN Brain Science Institute (Japan), the GMD-Japan Research Laboratory (Japan), Groningen University (The Netherlands), the Eindhoven University of Technology (The Netherlands), Starlab (Belgium), and the Bulgarian Academy of Science. She has been a project

leader of several multi-disciplinary projects, and coordinated the robotics and social robots educational and research team at ID department of TU/e. She is an associate editor of Journal of Integrative Neuroscience, and Personal and Ubiquitous Computing and has organized international conferences and has served as a program chair of IEEE and ACM conferences. She is an author of over 90 scientific papers, conference proceedings, and one book. She has expertise in modeling social behavior, social robotics, functional brain modeling for applications in robotics, on learning methods, and human centered interaction design.

Matthias Rauterberg received the B.S. in psychology (1978) at the University of Marburg (Germany), the B.S. in philosophy (1981) and computer science (1983), the M.S. in psychology (1981, summa cum laude) and computer science (1986, summa cum laude) at the University of Hamburg (Germany), and the Ph.D. in computer science/mathematics (1995, awarded) at the University of Zurich (Switzerland). He was a senior lecturer for 'usability engineering' in computer science and industrial engineering at the Swiss Federal Institute of Technology (ETH) in Zurich. He was the head of the Man–Machine Interaction Research Group (MMI) from the Department of Industrial Engineering at the ETH-Zurich (Switzerland). In 1998 he was appointed as fulltime professor for 'Human Communication Technology' and since 2012 he is professor for 'Interactive Systems Design', first at IPO—Center for Research on User-System Interaction, and later at the Department of Industrial Design at Eindhoven University of Technology (The Netherlands). From 1999 to 2001 he was director of the Center for Research on User-System Interaction (IPO), and director of the User-System-Interaction (USI) design program. He is now the head of the Designed Intelligence (DI) group. The DI group has a Simulation Lab, Robotics Lab, Biofeedback Lab, Game Lab, Cultural Computing Lab and an Interactive Vision Studio.

He was the Swiss representative in the International Federation for Information Processing (IFIP) Technical Committee (TC) 13 on 'Human Computer Interaction' (1994–2002) and the chairman of the IFIP Working Group (WG) 13.1 on 'HCI and Education' (1998–2004). He was the Dutch representative and the founding vice-chair of the IFIP TC14 on 'Entertainment Computing' (2002–2012). Since 2012 he is elected as chair of IFIP TC14 (term 2013–2015). He was also the chair of the IFIP WG14.3 on 'Entertainment Theory' (2004–2012). He was appointed as visiting professor at the distinguished private Kwansei Gakuin University (Japan) (2004–2007), and since 2011 he is guest professor at Jiangnan University (China) (2011–2013).

He received the prestigious German GI-HCI award for the best Ph.D. in 1997 and the Swiss Technology Award for the BUILD-IT system in 1998. Since 2004 he is a nominated member of the 'Cream of Science' in the Netherlands (the 200 top-level Dutch researchers) and amongst the 10 top-level TU/e scientists. Since 2007 he is holder of the Silver Core Award of the International Federation for Information Processing (IFIP).

He has over 400 publications in international journals, conference proceedings, books, etc. He is also editor and member of the editorial board of several leading international journals. He regularly acts as reviewer and evaluator for several national and international funding bodies.

His recent research is in the area of entertainment computing, cognitive systems, human–computer interaction, and design science.