Simulating Collaborative Learning Through Decision-Theoretic Agents

David V. Pynadath[1], Ning Wang[1]✉, and Richard Yang[2]

[1] Institute for Creative Technologies, University of Southern California
{pynadath, nwang}@ict.usc.edu
[2] Stanford University
richard.yang@cs.stanford.edu

Abstract. Simulation for team training has a long history of success in medical care and emergency response. In fields where individuals work together to make decisions and perform actions under extreme time pressure and risk (as in military teams), simulations offer safe and repeatable environments for teams to learn and practice without real-world consequences. In our team-based training simulation, we use intelligent agents to represent individual learners and to autonomously generate behavior while learning to perform a joint task. Our agents are built upon PsychSim, a social-simulation framework that uses decision theory to provide domain-independent, quantitative algorithms for representing and reasoning about uncertainty and conflicting goals. We present a collaborative learning testbed in which two PsychSim agents performed a joint "capture-the-flag" mission in the presence of an enemy agent. The testbed supports a reinforcement-learning capability that enables the agents to revise their decision-theoretic models based on their experiences in performing the target task. We can "train" these agents by having them repeatedly perform the task and refine their models through reinforcement learning. We can then "test" the agents by measuring their performance once their learning has converged to a final policy. Repeating this train-and-test cycle across different parameter settings (e.g., priority of individual vs. team goals) and learning configurations (e.g., train with the same teammate vs. train with different teammates) yields a reusable methodology for characterizing the learning outcomes and measuring the impact of such variations on training effectiveness.

Keywords: collaborative learning, team-based training, intelligent agent, reinforcement learning, social simulation

1 Introduction

A good team is more than a collection of individuals. In an effective team, each team member masters its individual role and coordinates with other team members to accomplish complex tasks. Good teams do not happen by accident. Team members train individually and together in order to do well as a team. Although team tasks are ubiquitous in today's society, team-based training, particularly with the use of simulations, has a long history in medical care, emergency responses, and the military (e.g., [10], [17], and [25]). Realistic simulations can offer safe and repeatable environments for teams to practice without the real-world consequences. However, simulations alone are often not enough to ensure learning. Instructional support is often needed to help the team and individuals in case of mistakes and impasse, and guide the team on the path to success. Instructional support in teams has its unique challenges, compared to such support delivered in individual learning settings. Decisions on the target (individual vs. team), channel (private vs. public), and timing of the feedback (immediate vs. delayed) and many more issues can greatly impact how such support is received by the team and the efficacy of the feedback [34]. The type of support and how and when it should be delivered depends on the team structure (e.g., with leadership or leaderless) and what the team is trying to learn (e.g., task-related vs. teamwork related, for review, see [7] and [27]). Mismatch between the support and the team needs can result in tutorial feedback being ignored at best and interfering with the team learning at worst [35].

Instead of testing with human participants, a simulation of how teams train together and how instructional feedback influences team members and a team is desired. Inspired by the challenge in the design of instructional support for team training, we have developed a testbed to simulate how team members learn together. In the current implementation of the testbed, team members are modeled as intelligent agents in a collaborative learning setting where they can learn from experience to improve team performance. Collaborative learning is often considered a type of team training, with emphasis on the team training how to collaborate to improve as a whole [27]. It is different from cooperative learning in that the agent does not try to maximize learning of other team members. However, our simulation testbed is not limited to collaborative learning only—each member of the team can learn to improve its own action, in addition to learning to collaborate with others, to improve team performance.

Instructional support in team tutoring can take many forms and often depends on the team structure. For example, tutorial feedback for a team with a vertical leadership structure is more likely to differ based on members at different levels. For a leaderless team, the feedback is likely to be structured more for peers [1]. When a team is actively engaged in learning, team members communicate among themselves to discuss best actions, ask each other questions, and explain their reasoning. In our simulation testbed, we build upon feedback from peers. Instead of receiving instructional support from a tutor, the simulated team members learn from their own experience and from each other.

In this paper, we present a reconfigurable testbed with three agents training in a joint capture-the-flag scenario. We propose a methodology by which the agents train through repeated practice of the task and refine their models through reinforcement learning. The agents then test their learning outcome by measuring the efficacy of a final policy. Repeating this train-and-test cycle across different parameter settings yields a reusable methodology for characterizing the learning outcomes and measuring the impact of such variations on training effectiveness. The testbed can thus serve as a sandbox to test instructional feedback and other alternative strategies of value in team-tutoring research.

2 Related Work

While there is a vibrant research community on automatically-generated instructional support for learning in an individual setting (for review, see [3]), research on such support in the context of team training is relatively scarce. Early research in team-based simulation focused on creating an environment that allows teams to practice together. The Advanced Embedded Training System (AETS) is one such effort [40]. AETS is an intelligent tutoring system built for an Air Defense Team on a ship's Combat Information Center to learn how to utilize the command and control system. While AETS enables multiple users to train as a team, assessment and feedback were given on an individual basis. Such feedback was then relayed to a human tutor, who offered team-based feedback. A similar effort is the Steve agent-based training simulation for emergency response on a military vessel [25]. In the training simulation, Steve agents can serve as a tutor as well as an individual team member, thus allowing the simulation to support a team of any combination of Steve agents and humans to train together. In the training simulation, Steve agents and humans learn to complete tasks through communication between team members.

More recently, there has been a resurgence of research into automated tutorial support for team training. One of the team training simulation testbeds implements a Multiple Errands Test, where a team of three completes errands following a shipping list in a virtual mall [35]. Using this testbed, a study on the influence of privacy (Public vs.Private) and audience (Direct vs. Group) of feedback showed no significant influences of such variables on team performance. A more recent effort is the Recon testbed that was built with the Generalized Intelligent Framework for Tutoring (GIFT) [7]. It supports the collaborative team task of reconnaissance [2]. Using the Recon testbed, researchers again experimented with variables in feedback to the teams, specifically target (individual vs. team), within 2-person teams [14].

Our testbed is used not for training but to simulate the training process. Agents learn to improve both their own and the team's performance from their own experience, by observing other agents, and by communicating with teammates. We draw upon the body of multiagent research on simulating teamwork and learning. Existing formalisms represent team goals, plans, and organizations that operationalize decision-making found in human teams [6, 9, 30]. Embedding these mechanisms within intelligent agents has enabled the construction of high-fidelity simulations of team behavior (e.g., simulated aircraft performing a joint mission [31]). The uncertainty and conflicting goals that are ubiquitous in most team settings led to decision-theoretic extensions of these models to incorporate quantitative probability and utility functions [32, 23]. More recently, agents have incorporated reinforcement learning (among other methods) to derive these models through experience and in a decentralized fashion, allowing individual agents to arrive at a coordinated strategy through experience [28, 19, 5].

3 PsychSim

We have built our testbed using the multiagent social simulation framework, PsychSim [15, 20]. PsychSim grew out of the prescriptive teamwork frameworks cited in Section 2 (especially [23]), but with a different aim toward being a descriptive model of human behavior. PsychSim represents people as autonomous agents that integrate two multiagent technologies: recursive models [8] and decision-theoretic reasoning [11]. Recursive modeling gives agents a Theory of Mind [39], to form complex attributions about others and incorporate such beliefs into their own behavior. Decision theory provides the agents with domain-independent algorithms for making decisions under uncertainty and in the face of conflicting objectives. We have used PsychSim to model a range of cognitive and affective biases in human decision-making and social behavior (e.g., [21, 22]).

Another motivation behind the use of PsychSim is its successful application within multiple simulation-based learning environments. The Tactical Language Training System (TLTS) is an interactive narrative environment in which students practice their language and culture skills by talking to non-player characters built upon PsychSim agents [26]. We also used PsychSim's mental models and quantitative decision-theoretic reasoning to model a spectrum of negotiation styles within the ELECT BiLAT training system [12]. Additionally, UrbanSim used a PsychSim-driven simulation to put trainees into the role of a battalion commander undertaking an urban stabilization operation [16]. In SOLVE, PsychSim agents populate a virtual social scene where people could practice techniques for avoiding risky behavior [13, 18].

We have also used PsychSim to build experimental testbeds for studying human teamwork. In one such testbed, we used a PsychSim agent to autonomously generate behaviors for a simulated robot that teamed with a person, in a study of trust within human-robot interaction [38, 36]. Another PsychSim-based testbed gave four human participants a joint objective of defeating a common enemy, but with individual scores that provided some impetus for competitive behavior within the ostensible team setting [24]. We build upon PsychSim's capability for such experimental use in the expanded interaction of the current investigation.

4 Team-based Training Simulation

In our testbed, we implement a "capture-the-flag" scenario. In the scenario, a team of trainees learn how to work together to attack a goal location being defended by a team of enemies. Both the trainees and enemies are represented as PsychSim agents. In the preliminary testing described here, the blue team consists of three agents, while the red team consists of only one (denoted as *Enemy*). The three blue agents are assigned to three distinct roles: the *Attacker* tries to reach the goal location, the *Decoy* tries to lure the enemy away from the *Attacker*, and the *Base* decides whether or not to deploy the *Decoy*.

Ideally, the *Attacker* should proceed to the goal while maintaining a safe distance from the *Enemy*. If the *Enemy* detects the *Attacker* and approaches it, the *Base* should deploy the *Decoy*. The *Decoy* should then approach the *Enemy* to draw its attention away from the *Attacker*. Such a coordinated strategy will maximize the chance that the team achieves its objective, while minimizing the chance that the *Enemy* captures any team members (see Figure 1).

PsychSim represents the decision-making problem facing the agents as a Partially Observable Markov Decision Process (POMDP) [11]. Partial observability accounts for the fact that the agents cannot read each other's minds and that they may have incomplete or noisy observations of the environment. However, in this presentation, we make the environment itself completely observable, reducing the domain to a Markov Decision Process (MDP) instead. An MDP is a tuple $\langle S, A, P, R\rangle$, with S being the set of states, A the set of actions, P the transition probability representing the effects of the actions on the states, and R the reward function that expresses the player's preferences.

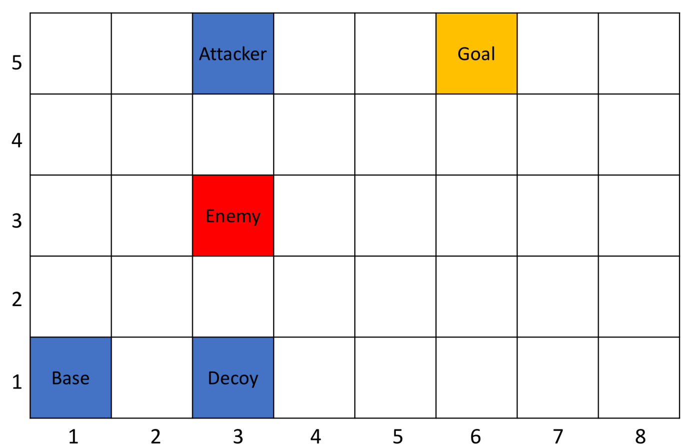

Fig. 1. A mid-mission screenshot of the "capture-the-flag" scenario. The *Attacker*, *Base* and *Decoy* are located at [3,5], [1,1] and [3,1], while the *Enemy* and the goal are located at [3,3] and [6,5].

The state of the world, S, represents the evolution of the game state over time. We use a factored representation [4] that allows us to separate the overall game state into orthogonal features that are easier to specify and model. The locations of the agents and of the goal are specified by x and y coordinates on a grid. The grid is 5×8 in the specific configuration described here, but obviously other grid sizes are possible. There is also a cost associated with deploying the

Decoy agent, as opposed to letting the other agent go solo. The actions, A, available to the *Attacker*, *Decoy*, and *Enemy* agents are moves in one of the four directions or waiting in their current location. The *Base* can either deploy the *Decoy* agent or wait. The transition probability, P, represents the effect of the agents' movement decisions, which we specify here to succeed with 100% reliability. In general, the P function can capture any desired stochastic error (e.g., due to terrain or visual conditions).

The *Attacker* agent has two potentially conflicting objectives within its reward function, R: minimizing its distance to the goal (i.e., to try and reach the goal) and maximizing its distance from the *Enemy* (i.e., to avoid capture). More precisely, the *Attacker*'s reward function is a weighted sum of the difference between its x and y values and the goal's and between its x and y values and the *Enemy*'s. The *Decoy* agent also has two potentially conflicting objectives: minimizing its distance from *Enemy* and maximizing the distance between the *Attacker* and *Enemy*. It thus tries to lure the *Enemy* toward itself and away from the *Attacker*. The *Base* agent's conflicting objectives consist of also minimizing the distance between the *Decoy* and *Enemy*, while also minimizing the cost of deploying the *Decoy*. Finally, the *Enemy* agent seeks to minimize its distance to the *Attacker* and *Decoy* agents (i.e., to capture them if possible, or at least drive them away). Thus, each agent has two conflicting objectives within its reward function, and the weights assigned to each determine their relative priority. Modifying these weights will change the incentives that each agent perceives.

Having specified the game within the PsychSim language, we can apply existing algorithms to autonomously generate decisions for individual agents [11]. Such algorithms enable the agent to consider possible moves (both immediate and future), generate expectations of the responses of the other agents, and compute an expected reward gain (or potentially loss) for each such move. It then chooses the move that maximizes this expected reward. Importantly, this algorithm can autonomously generate behavior without any additional specification, allowing us to observe differences in behavior that result from varying modeling parameters (e.g., the relative priority between objectives).

5 Evaluation

To evaluate the testbed's suitability for studying collaborative learning, we simulated the scenario with alternate configurations of the *Attacker* agent to explore the space of team behavior and outcomes. Our goal is to verify that varying the agent's model (especially its reward function) will lead to different individual behaviors and team outcomes and uncover what and how the team should train to improve. To quantify the team outcome, the blue team is given a score that is a weighted sum of the distance between *Attacker* and the goal (0 means success), distance between *Attacker* and *Enemy* (0 means capture and immediate failure), the cost incurred from *Decoy* deployment, and the duration of the task as a function of total number of turns. During the experiment, each mission has a maximum duration of 20 turns, as that length was generally sufficient for a specific configuration to succeed if it ever would. Missions where the *Attacker* reached the goal in fewer than 10 turns were given a bonus score. Figure 2 shows the overall team score (blue means better, red means worse) as a function of the *Attacker*'s reward weights. The X axis represents the weight of getting closer to the goal, while the Y axis represents the weight of getting closer to the *Enemy*. In other words, in the right (left) half of the graph, it wants to move toward (away from) the goal, and in the bottom (top) half, it wants to move away from (toward) the enemy.

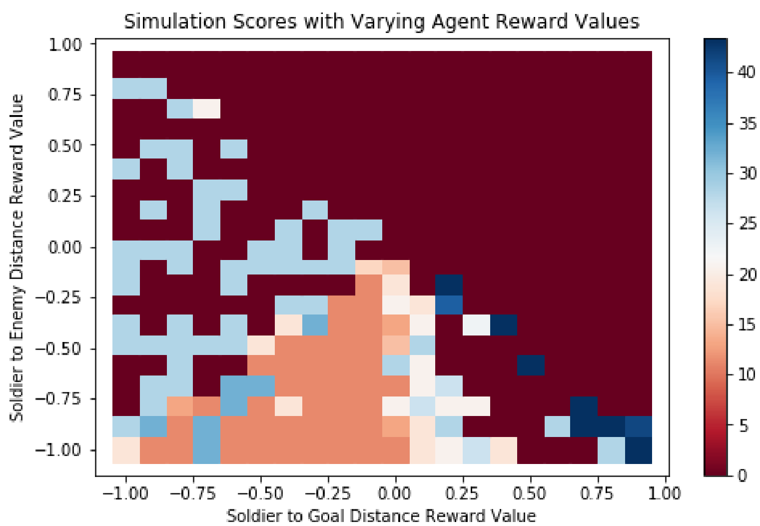

Fig. 2. Blue team's overall performance as a function of *Attacker* reward weights

Not surprisingly, the team's top performance is in the bottom right, where the *Attacker* minimizes its distance to the goal and maximizes its distance to the *Enemy*—i.e., it tries to reach the goal while avoiding capture. The success at point (-1,1) gives equal weight to the two objectives of team actions, but we can see that the team can achieve similarly high performance at other weightings along the diagonal in the bottom-right region. This balance is a function of our scoring metric that gave equal weight (in magnitude) to those two outcomes.

We can also see where the blue team needs to improve by learning a better balance (i.e. the reward weights) of its objectives. In particular, there is a large light-blue region of positive results on the left of the graph, i.e., where *Attacker* instead carries out actions to maximize distance from the goal. By staying away from the goal, the agent also generally stays away from the *Enemy*, who starts off near the goal. Thus, capture is very rare in this region, but mission success is also rare. This region provides a challenge for the team's training, which must ensure that the *Attacker* agents who start off in this light-blue region move through the intervening light-red regions (where they will achieve bad outcomes) to get to the superior, but relatively hard-to-find, dark blue points in the bottom right.

6 Discussion

The existing testbed thus provides an interesting space of team behaviors, even within this small-scale configuration. By representing this scenario on top of a general multiagent framework, we gain access to a wide space of possible reconfiguration dimensions that can be used for future investigations. In this section, we propose a series of such reconfigurations that would be valuable for studying collaborative learning and team training. For example, the testbed provides a challenging environment for reinforcement learning, where individual trainees learn from their own experience to balance their objectives. We can incorporate reinforcement learning into our PsychSim agents to simulate how each teammate can improve its behavior through its own experience [29]. Using model-based reinforcement learning, the agents can change the weights within their reward function based on the outcomes of their decisions. For example, if the *Attacker* gets captured, it will increase the weight associated with moving away from the enemy. If it does not get captured, but fails to reach the goal, it will increase the weight associated with nearing the goal. Such a procedure will allow the *Attacker* to dynamically learn a reward function that is optimized with respect to mission objectives.

However, the *Decoy* and *Base* agents receive less direct feedback for their decisions. We can instead allow them to learn by observing the outcomes for their *Attacker* teammate. For example, if the goal is not achieved even after avoiding capture, the *Decoy* could give a higher weight to drawing the *Enemy* to itself. Alternatively, it could introduce a new objective of minimizing the distance between the *Attacker* and goal, giving the *Decoy* an explicit model of the goal objective. By updating these three weights, we can explore the ability for the *Attacker*'s teammates to learn from its direct feedback. We can thus vary the feedback (i.e., the reinforcement learning signal) received by the agents in terms of the credit and blame for outcomes. Alternatively, we can broadcast the feedback to the entire team, causing the agents to update their models of their teammates as well using PsychSim's Theory of Mind capability. In general, this mechanism allows us to experiment with different feedback signals to give individual team members based on mission outcomes and team learning.

One key advantage of using an agent framework like PsychSim is that we have many dimensions along which we can enrich the reasoning of our learners. For example, in the current configuration, all of the agents know each others' objectives. This is not a realistic model of human teamwork, where people rarely know exactly how important team vs. individual objectives are to their teammates. Fortunately, PsychSim's Theory of Mind reasoning allows us to easily give the agents uncertainty about the reward function of other agents. We can thus expand our agents' learned behaviors to consider not just the locations of their teammates, but also their subjective perspectives.

Introducing uncertainty also necessitates communication among teammates. Successful teamwork uses communication to maintain shared situational awareness about task progress, teammate status, etc. [6, 9, 30]. We can leverage our underlying agent architecture's existing algorithms for belief update [11] and communication [15] to explore alternate communication strategies to establish coherent joint beliefs among team members. In other words, our learning agents would expand their action space to include possible messages, such as "There is a 90% chance that the *Enemy* is at (3,3)". They would subsequently arrive at a learned behavior that specifies the best conditions under which to send such messages (e.g., if no one has found the enemy yet, then report your estimated location of the *Enemy* when your confidence is $> 75\%$).

We can reuse this mechanism to explore the effect of post-mission communication as well. Upon learning to maximize their individual performance, agents can communicate their learned policy to other team members, particularly those still performing suboptimally. Such communication would simulate a form of peer tutoring [33] commonly seen in collaborative learning. We could also enrich this communication to include an agent's explanation of its optimal policy (e.g., using [37]) to justify its choice to its teammates. We can also investigate alternate channels for this team communication, for example, allowing messages addressed to an individual agent vs. messages broadcast to the whole team.

Once our agents are learning about teammates, we can use our testbed to study different team training configurations. For example, we could let a team of agents "train" by repeating missions until they learn a good coordination policy. Then, we could "test" the team by replacing a team member with an agent that had not performed any learning. Alternatively, we could have each agent train separately with continually changing team members, and then test test a team of agents that have trained in such a fashion. By quantifying the performance outcomes of these different training methods under different task and environment configurations, we can gain potential insight into the conditions under which each can be expected to improve team performance. For example, we could measure the benefit of introducing an "experienced" team member (an agent who has learned about the domain in prior iterations) into an "inexperienced" team (agents who have never operated in the domain before). Simulating the performance of such a team might (for example) show that the experienced agent provides a "tutoring" benefit when post-mission communication is allowed to support learning, but can actually hinder performance (because of expectation mismatches) without such communication.

While the work discussed here focuses on simulations of how team trains together with virtual agents, it can help inform the design of intelligent team tutoring systems for real human teams. For example, one of the decisions an intelligent tutor needs to make is when to provide the feedback. Immediate feedback may help the team on the task at hand but interfere with team building. Delayed feedback may result in frustration after the team exhausts options and fails. Outcomes from simulations of tutorial feedback given at different times (immediate vs. delayed vs. a combination of the two) can help the designers of such intelligent tutors weigh the trade-offs between the choices of timing. Additionally, using PsychSim agents, we can simulate teams made up of members of varied characteristics, e.g., prior knowledge and motivation, and experiment with how decisions on tutorial feedback, such as target, channel and timing, impact the team's learning. In conclusion, the multiagent testbed we have constructed uses a relatively simple coordination scenario as a jumping-off point for a wide variety of potential simulations of collaborative learning and team training that can have implications for intelligent tutoring systems for real-human teams.

7 Acknowledgment

This project is funded by the U.S. Army Research Laboratory. Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

References

1. Bonner, D., Gilbert, S., Dorneich, M.C., Burke, S., Walton, J., Ray, C., Winer, E.: Taxonomy of teams, team tasks, and tutors. In: GIFT Users Symposium. p. 189 (2015)
2. Bonner, D., Walton, J., Dorneich, M.C., Gilbert, S.B., Sottilare, R.A.: The development of a testbed to assess an intelligent tutoring system for teams. In: AIED Workshop on Developing a GIFT (2015)
3. du Boulay, B.: Recent meta-reviews and meta–analyses of aied systems. IJAIED 26(1), 536–537 (2016)
4. Boutilier, C., Dean, T., Hanks, S.: Decision-theoretic planning: Structural assumptions and computational leverage. JAIR 11(1), 94 (1999)
5. Busoniu, L., Babuska, R., De Schutter, B.: A comprehensive survey of multiagent reinforcement learning. IEEE Transactions on SMC 38(2), 156–172 (2008)
6. Cohen, P.R., Levesque, H.J.: Teamwork. Nous 25(4), 487–512 (1991)
7. Gilbert, S.B., Slavina, A., Dorneich, M.C., Sinatra, A.M., Bonner, D., Johnston, J., Holub, J., MacAllister, A., Winer, E.: Creating a team tutor using gift. IJAIED pp. 1–28 (2017)
8. Gmytrasiewicz, P.J., Durfee, E.H.: A rigorous, operational formalization of recursive modeling. In: ICMAS. pp. 125–132 (1995)
9. Grosz, B.J., Kraus, S.: Collaborative plans for complex group action. AIJ 86(2), 269–357 (1996)
10. Heinrichs, W.L., Youngblood, P., Harter, P.M., Dev, P.: Simulation for team training and assessment: case studies of online training with virtual worlds. World Journal of Surgery 32(2), 161–170 (2008)
11. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. AIJ 101(1), 99–134 (1998)
12. Kim, J.M., Hill, Jr., R.W., Durlach, P.J., Lane, H.C., Forbell, E., Core, M., Marsella, S., Pynadath, D., Hart, J.: BiLAT: a game-based environment for practicing negotiation in a cultural context. IJAIED 19(3), 289–308 (2009)

13. Klatt, J., Marsella, S., Krämer, N.C.: Negotiations in the context of aids prevention: an agent-based model using theory of mind. In: IVA. pp. 209–215 (2011)
14. MacAllister, A., Kohl, A., Gilbert, S., Winer, E., Dorneich, M., Bonner, D., Slavina, A.: Analysis of team tutoring training data. In: MODSIM World (2017)
15. Marsella, S.C., Pynadath, D.V., Read, S.J.: PsychSim: Agent-based modeling of social interactions and influence. In: ICCM. pp. 243–248 (2004)
16. McAlinden, R., Pynadath, D., Hill, Jr., R.W.: UrbanSim: Using social simulation to train for stability operations. In: Ehlschlaeger, C. (ed.) Understanding Megacities with the Reconnaissance, Surveillance, and Intelligence Paradigm, pp. 90–99 (2014)
17. Merién, A., Van de Ven, J., Mol, B., Houterman, S., Oei, S.: Multidisciplinary team training in a simulation setting for acute obstetric emergencies: a systematic review. Obstetrics & Gynecology 115(5), 1021–1031 (2010)
18. Miller, L.C., Marsella, S., Dey, T., Appleby, P.R., Christensen, J.L., Klatt, J., Read, S.J.: Socially optimized learning in virtual environments (SOLVE). In: ICIDS. pp. 182–192 (2011)
19. Panait, L., Luke, S.: Cooperative multi-agent learning: The state of the art. JAAMAS 11(3), 387–434 (2005)
20. Pynadath, D.V., Marsella, S.C.: PsychSim: Modeling theory of mind with decision-theoretic agents. In: IJCAI. pp. 1181–1186 (2005)
21. Pynadath, D.V., Marsella, S.C.: Socio-cultural modeling through decision-theoretic agents with theory of mind. In: Nicholson, D.M., Schmorrow, D.D. (eds.) Advances in Design for Cross-Cultural Activities, pp. 417–426. CRC Press (2013)
22. Pynadath, D.V., Si, M., Marsella, S.C.: Modeling theory of mind and cognitive appraisal with decision-theoretic agents. In: Gratch, J., Marsella, S. (eds.) Social emotions in nature and artifact: Emotions in human and human-computer interaction, chap. 5, pp. 70–87. Oxford University Press (2014)
23. Pynadath, D.V., Tambe, M.: The communicative multiagent team decision problem: Analyzing teamwork theories and models. JAIR 16, 389–423 (2002)
24. Pynadath, D.V., Wang, N., Merchant, C.: Toward acquiring a human behavior model of competition vs. cooperation. In: I/ITSEC (2015)
25. Rickel, J., Johnson, W.L.: Virtual humans for team training in virtual reality. In: AIED. p. 585 (1999)
26. Si, M., Marsella, S.C., Pynadath, D.V.: Thespian: Using multi-agent fitting to craft interactive drama. In: AAMAS. pp. 21–28 (2005)
27. Sottilare, R.A., Burke, C.S., Salas, E., Sinatra, A.M., Johnston, J.H., Gilbert, S.B.: Designing adaptive instruction for teams: A meta-analysis. IJAIED pp. 1–40 (2017)
28. Stone, P., Veloso, M.: Multiagent systems: A survey from a machine learning perspective. Autonomous Robots 8(3), 345–383 (2000)
29. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT Press (1998)
30. Tambe, M.: Towards flexible teamwork. JAIR 7, 83–124 (1997)
31. Tambe, M., Johnson, W.L., Jones, R.M., Koss, F., Laird, J.E., Rosenbloom, P.S., Schwamb, K.: Intelligent agents for interactive simulation environments. AI Magazine 16(1), 15–40 (1995)
32. Tambe, M., Zhang, W.: Towards flexible teamwork in persistent teams. JAAMAS 3(2), 159–183 (2000)
33. Topping, K.J.: The effectiveness of peer tutoring in further and higher education: A typology and review of the literature. Higher Education 32(3), 321–345 (1996)
34. Walton, J., Dorneich, M.C., Gilbert, S., Bonner, D., Winer, E., Ray, C.: Modality and timing of team feedback: Implications for GIFT. In: GIFT Users Symposium. pp. 190–198 (2014)

35. Walton, J., Gilbert, S.B., Winer, E., Dorneich, M.C., Bonner, D.: Evaluating distributed teams with the team multiple errands test. In: I/ITSEC (2015)
36. Wang, N., Pynadath, D.V., Hill, S.G.: Building trust in a human-robot team. In: I/ITSEC (2015)
37. Wang, N., Pynadath, D.V., Hill, S.G.: The impact of POMDP-generated explanations on trust and performance in human-robot teams. In: AAMAS (2016)
38. Wang, N., Pynadath, D.V., Shankar, S., K.V., U., Merchant, C.: Intelligent agents for virtual simulation of human-robot interaction. In: HCI. pp. 228–329 (2015)
39. Whiten, A., Byrne, R.: Natural theories of mind: Evolution, development and simulation of everyday mindreading. B. Blackwell Oxford, UK (1991)
40. Zachary, W., Cannon-Bowers, J.A., Bilazarian, P., Krecker, D.K., Lardieri, P.J., Burns, J.: The advanced embedded training system (AETS): An intelligent embedded tutoring system for tactical team training. IJAIED 10, 257–277 (1998)