Adaptive Robot Assisted Therapy
Using Interactive Reinforcement Learning

Konstantinos Tsiakas[1,2(✉)], Maria Dagioglou[2],
Vangelis Karkaletsis[2], and Fillia Makedon[1]

[1] Computer Science and Engineering Department,
University of Texas at Arlington, Arlington, USA
konstantinos.tsiakas@mavs.uta.edu, makedon@uta.edu
[2] National Center for Scientific Research "Demokritos", Athens, Greece
{mdagiogl,vangelis}@iit.demokritos.gr

Abstract. In this paper, we present an interactive learning and adaptation framework that facilitates the adaptation of an interactive agent to a new user. We argue that Interactive Reinforcement Learning methods can be utilized and integrated to the adaptation mechanism, enabling the agent to refine its learned policy in order to cope with different users. We illustrate our framework with a use case in the domain of Robot Assisted Therapy. We present our results of the learning and adaptation experiments against different simulated users, showing the motivation of our work and discussing future directions towards the definition and implementation of our proposed framework.

1 Introduction

An interactive learning agent is an entity that learns through the continuous interaction with its environment. Such agents act based on the information they perceive and their own policy. This interaction can be seen as a stochastic sequential decision making problem, where the agent must learn a policy that dictates what to do at each interaction step, by accumulating knowledge through experience [9].

Reinforcement Learning (RL) provides an appropriate framework for interaction modeling and optimization of problem that can be formulated as Markov Decision Processes (MDP) [17]. RL methods have been successfully applied for modeling the interaction in problems as Adaptive Dialogue Systems, Intelligent Tutoring Systems and recently to Robot Assisted Therapy applications [7,16]. The advantage of RL methods in modeling the interaction in such systems is that the stochastic variation in user responses is depicted as transition probabilities between states and actions [19].

In real-world systems, where the state-action space is large and the environment is dynamic, learning an optimal policy for each specific user is challenging. An interactive agent designed to interact with different users should be able to adapt to environmental changes by efficiently modifying a learned policy to cope with different users, instead of learning from scratch for each user. Furthermore,

even the same user may not be consistent over time, in terms of reactions and intentions. In addition, user abilities change over time, as users adapt their own behavior while interacting with a learning agent. Recent works investigate the aspect of *co-adaptation* in man-machine interaction systems, assuming that both agent and user adapt in a cooperative manner to achieve a common goal [10].

Taking these points into consideration, a dynamic adaptation mechanism is required for an interactive agent that enables the system to adapt to different users, ensuring efficient interactions. In this paper, we present an interactive learning and adaptation framework for Robot Assisted Therapy, that utilizes *Interactive Reinforcement Learning* methods to facilitate the policy adaptation of the robot towards new users. We argue that interactive learning techniques [3,14] can be combined with transfer learning methods [8] to facilitate the adaptation of an agent to a new user.

2   Related Work

Robot Assisted Therapy (RAT) has been extensively applied to assist users with cognitive and physical impairments [18]. An assistive robot should be able to adapt to user behavior preferences and changes, ensuring a safe and tailored interaction [11]. Such systems may also support multiparty interactions [12], including a *secondary user* (therapist, clinician) who supervises the interaction between the *primary user* and the agent.

There are works that indicate that personalized and tailored robotic assistive systems can establish a productive interaction with the user, improving the effects of a therapy session. In [15], the authors present a therapeutic *robocat* to investigate how patients with dementia would respond. Their interactions with the robocat led to less agitation, and more positive experiences. Similarly, in [25], the authors present Paro, a robotic seal that interacted proactively and reactively with patients, showing that the interaction with Paro had psychological, physiological, and social effects on elderly people. In [21], the authors proposed an adaptive socially assistive robotic (SAR) system that provides a customized protocol through motivation, encouragement and companionship for users suffering from Alzheimer's disease.

Our work moves towards the definition and implementation of an interactive learning and adaptation framework for interactive agents [23]. Based on this framework, an interactive agent is able to refine a learned policy towards a new user, by exploiting additional communication channels (feedback and guidance) provided by the users during the interaction.

3   Adaptive Robot Assisted Therapy

In this section, we present an adaptive training task example, where the user needs to perform a set of cognitive or physical tasks. The representation we present is applicable to various RAT applications, as rehabilitation exercises and cognitive tasks [1,21,24]. We follow a scenario where the user interacts with the

robot during a training session. The user must complete a set of three predefined tasks. Each task has four difficulty levels (Easy, Medium, Normal, Hard). The robot keeps track of task duration and the user's score. To measure user's score, we follow a negative marking approach. At each interaction step, the user receives a positive score proportional to the task difficulty, upon a successful turn and the corresponding negative one for failure. The robot keeps track of these scores and sums them to compute the user's total performance.

3.1   Robot Behavior Modeling

In order to model the robot's behavior, we formulate the problem as an MDP. The state is a set of variables that represent the current task state: Task ID (1,2,3), Task Duration (0–6), Difficulty Level (1–4) and Score (−4–4). At each interaction step, the system selects a difficulty level for the current task and the user performs the task. The agent receives a reward, proportional to the user's score, to update its policy (Fig. 1).
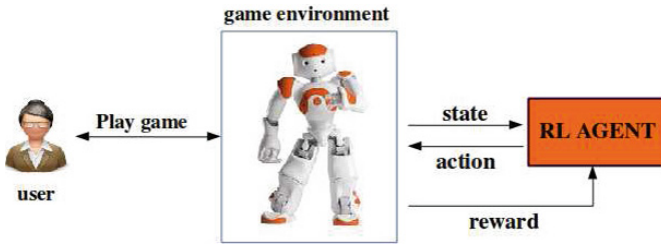

Fig. 1. The task formulated as a Reinforcement Learning problem. The robot keeps track of the current state. The RL agent selects an action (the next difficulty level or task switching) and the robot plays with the user, returning a reward to the agent to update its policy.

The agent needs to learn the optimal policy; the mapping from states to actions that maximizes the accumulated reward (or total return) during each interaction. Maximizing total return can be translated as providing the appropriate difficulty level that will maximize user's performance.

In order to evaluate our robot behavior modeling, we defined four different user models that capture different user abilities. These user models depict the user skills under different game parameters (task difficulty and duration). Each user model is a rule-based model whose binary output indicates user's success (or failure) for each task difficulty and duration parameters. In Fig. 2, we show two of the defined user models that capture different user skills. The agent must learn how to adjust the difficulty level and when to switch to the next task, for each user model, learning a *user-specific policy (USP)*.

User 1 (expert)		Difficulty level			
		easy	medium	normal	hard
d	0	1	1	1	1
u	1	1	1	1	1
r	2	1	1	1	1
a	3	1	1	1	1
t					
i	4	1	1	1	1
o	5	1	1	1	1
n	6	0	0	0	0

User 2 (novice)		Difficulty level			
		easy	medium	normal	hard
d	0	1	1	1	1
u	1	1	1	1	1
r	2	1	1	0	0
a	3	1	0	0	0
t					
i	4	1	0	0	0
o	5	1	0	0	0
n	6	0	0	0	0

Fig. 2. User model examples. User 1 is an expert user, able to complete all tasks at the highest difficulty level. User 2 is a novice user that can perform each task at the high level only for the first two rounds and the rest of them at a lower difficulty. Since both users do not succeed after time duration 6 (maximum), the agent must learn to switch to the next task.

3.2   Learning Experiments

Our first step is to train the agent against the four different user models. For our learning experiments, we applied the Q-learning algorithm, following the ϵ-greedy policy with linearly decreasing exploration rate. An episode is a complete game of three tasks. In Fig. 3, we show the learning results for each user model.

We visualize the evaluation metrics, as proposed by [2]. We plot the total return collected during each episode, averaged over a number of episodes (1 epoch = 50 episodes). Since the agent must learn the best policy for each user so as to maximize their performance, we observe the similarity of total return and performance curves. Another metric is the start state value, which provides an estimate of the expected total return the agent can obtain by following this policy. In the top right figure, we observe the different convergence points for each user model. Since start state value expresses the expected return, we observe that start state value and average return tend to approximate each other as training evolves. Another convergence evaluation metric is the Q-value updates of all action-state pairs per epoch, showing that the algorithm converges as it learns, decreasing the state value updates.

3.3   Policy Transfer Experiments

Our next step is to evaluate these four USP policies to the different user models. We make two hypotheses: (1) a user specific policy is the optimal policy (for the corresponding model); the one that maximizes total return, thus user performance, and (2) applying a learned policy to a different user model may not be efficient but better than learning from scratch. We applied the four different USP to the four different user models, following an exploitation-only approach, since following an exploration strategy may not be safe for real-world HRI applications.
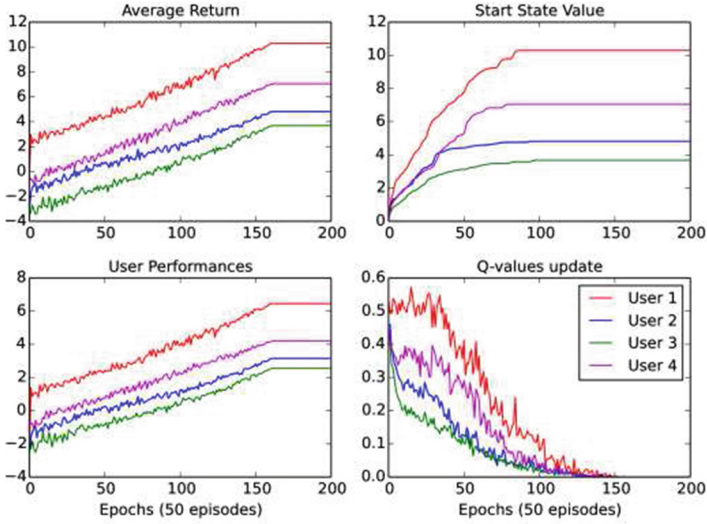
Fig. 3. Learning experiments. Applying Q-learning for the different user models results to different *user-specific policies (USP)*.
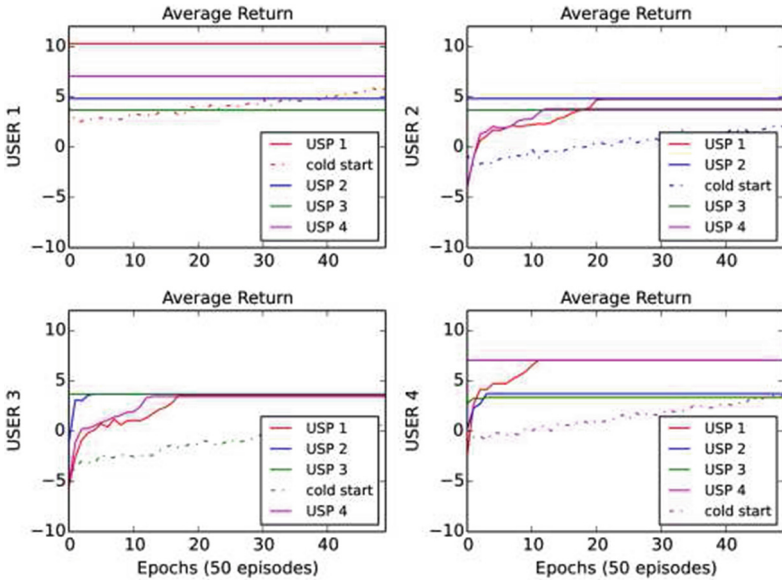

Fig. 4. Policy transfer experiments. In this experiment, we applied all learned USP to the different user models.

These policy transfer experiments validate our two hypotheses (Fig. 4). Each USP is the optimal policy for the corresponding model; it maximizes the total return. Moreover, applying a policy to a different user model may not be efficient but better than learning from scratch (dashed-line). We can observe three cases: (1) the initial policy adapts and converges to the USP, (2) the initially policy is improved but does not converge to the USP and (3) the learned policy remains unchanged. On the bottom right figure, we observe that all USPs adapt and converge to USP4. This happens because the agent interacts with different user models, receives negative rewards for specific state-action pairs, improving its policy for the corresponding pairs. However, on the top left figure, we observe that all policies remain unchanged. This happens because all policies result to positive rewards, since user-1 succeeds in all difficulty levels. However, only USP1 is the optimal policy for this user model. This indicates a need for a dynamical adaptation mechanism that enables the agent to efficiently refine its policy towards a new user.

4   Interactive Learning and Adaptation Framework

In this section, we present an interactive learning and adaptation framework that integrates Interactive Reinforcement Learning approaches to the adaptation mechanism. *Interactive Reinforcement Learning* (IRL) is a variation of RL that studies how a human can be included in the agent learning process. Human input can be either in the form of feedback or guidance. *Learning from Feedback* treats the human input as a reinforcement signal after the executed action [14]. *Learning from Guidance* allows human intervention to the selected action before execution, proposing (corrective) actions [8]. To our knowledge, IRL methods have not been investigated for the adaptation of an agent to a new environment. Hence, we propose their integration to the adaptation mechanism, as *policy evaluation* metrics used to evaluate and modify a learned policy towards an optimal one, following proper transfer methods.

Based on this framework (Fig. 5), an interactive agent is able to adapt a learned policy towards a new user by exploiting additional communication channels (feedback and guidance), provided during the interaction. Our framework supports the participation of a secondary user who supervises the interaction in its early steps, avoiding unsafe interactions. The supervisor can either physically or remotely supervise the interaction (*observation*). A user interface can be used to provide the supervisor with useful information about the agent learning procedure (*agent data*) to help them monitor the interaction and enhance their own decision making, before altering the agent's policy. The goal of this framework is to enable agents to learn as long as they interact with primary and secondary users, adapting and refining their policy dynamically.

User feedback can be considered as a *personalization* factor, as it is provided by the primary user *implicitly* (e.g., facial expressions, eye-tracking, engagement levels, etc.) and can be used to evaluate the interaction, thus the agent policy. In our adaptive training game case, one way to evaluate the agent policy is
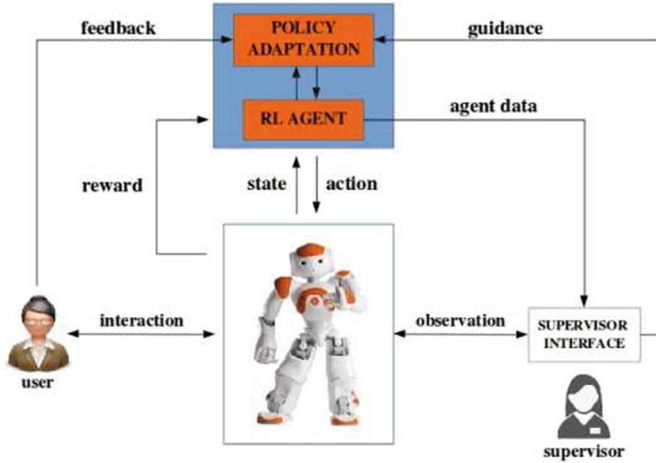
Fig. 5. We extend the RL framework by adding two additional communication channels; *feedback* and *guidance*. Their integration to the adaptation module can enable the agent to continuously adapt towards the current user, ensuring a safe and personalized interaction

to measure user engagement. We propose to use the Muse EEG headset[1], a commercially available tool that measures electrical activity at the scalp as it relates to various cognitive and mood states. This type of sensor can be used to measure how engaged a person is while that person is completing some sort of games or music tasks [13]. When the selected difficulty level is lower than needed, then the user may not be engaged. Moreover, when the task is difficult enough, the user may be frustrated and disengaged [4]. This implicit feedback can be exploited and efficiently modify a learned policy towards a new user, in an online and dynamic fashion.

On the other hand, guidance can be considered as a *safety* factor, following a supervised progressively autonomous approach [20], but also integrating human advice to the adaptation mechanism, enabling the agent adapt more effectively. The therapist, as a secondary user, can also set their own therapeutic goals by altering the policy. Making the learning process transparent to the secondary user may result to more informative guidance [5]. Informative metrics as state uncertainty and importance can be utilized to assist the secondary user provide the system with valuable guidance, in the form of corrective or suggested actions. Additionally, Active Learning methods [6] can be used to learn, based on state information, when the therapist should intervene, minimizing the expert's workload as the system learns.

---

[1] http://www.choosemuse.com/.

4.1   Preliminary Experiments

In this section, we present our preliminary adaptation experiments, following the proposed framework. As we mentioned, we assume that user feedback (engagement level) relates to the difficulty level variance; if the robot selects the appropriate difficulty the engagement should be high, thus the feedback value. For our simulation experiments, feedback is the normalised absolute difference between the selected and the appropriate difficulty, so as $feedback \in [-1,0]$. The feedback is used to modify the Q-values, following the Q-augmentation technique [14].

On the other hand, we use guidance in the form of corrective actions, following a semi-supervised autonomy approach [20] combined with *teaching on a budget* [22]. Based on this approach, the agent proposes an action based on its policy. The therapist can reject this action and select another, for a limited number of interventions ($M = 2$). For our experiments, the corrective actions are selected based on the corresponding USP with probability 0.8, to cover possible therapist errors. In Fig. 6, we show the results of our experiments, integrating feedback and guidance. We observe that for all cases, the integration of feedback and guidance improve the applied policy, resulting to its convergence to the corresponding USP (optimal policy), validating our hypothesis that interactive learning methods can be utilized for the policy transfer and adaptation.
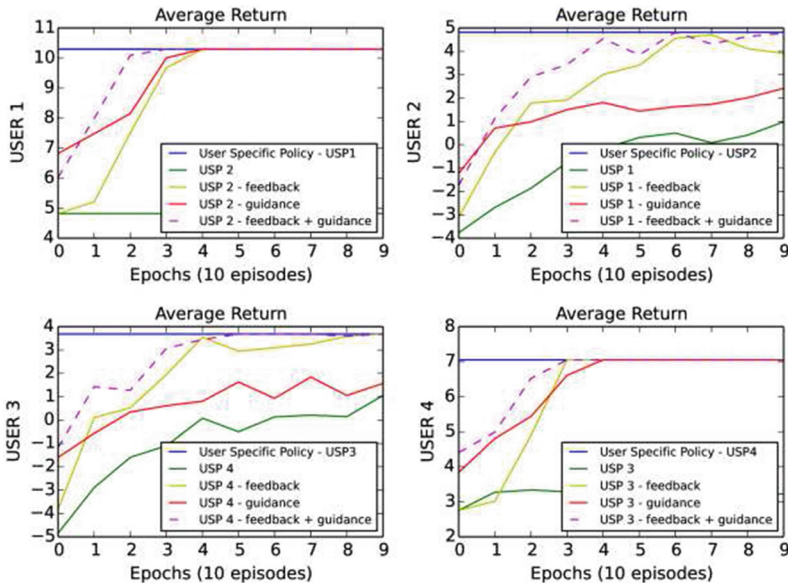

Fig. 6. Integrating feedback and guidance. We apply a different policy to each user model (a) without feedback/guidance, (b) with feedback and (c) with feedback and guidance. We observe that their integration to the learning improves the learned policy.

5   Discussion and Future Work

To conclude with, we introduced an interactive learning and adaptation framework for dynamically adaptive agents. We presented our use case in Robot Assisted Therapy with an adaptive training task. Our preliminary learning and adaptation experiments indicate that interactive learning methods can be integrated to the adaptation mechanism, resulting to an intelligent adaptive robot behavior.

Our preliminary results are promising, since the integration of interactive learning techniques facilitate the policy adaptation. However, there are some limitations on the presented framework and simulation experiments. Based on the defined user models, users are consistent over time and towards all tasks. This is likely to be violated in a real-world scenario. However, we believe that the proposed framework can be evolved and applied to real-world HRI applications, investigating further how interactive learning techniques can be integrated to the adaptation mechanism, considering user inconsistency over time and co-adaptation. Moreover, we will investigate how users (both primary and secondary) interact under this framework, developing appropriate interaction techniques. Our next steps include the implementation of a RAT scenario following the framework and a case study with participants to evaluate the task itself, as well as the proposed framework.

References

1. Andrade, K.d.O., Fernandes, G., Caurin, G.A., et al.: Dynamic player modelling in serious games applied to rehabilitation robotics. In: Robotics Symposium and Robocontrol, pp. 211–216. IEEE (2014)
2. Bellemare, M.G., Naddaf, Y., Veness, J., Bowling, M.: The arcade learning environment: an evaluation platform for general agents. J. Artif. Intell. Res. **47**, 253–279 (2012)
3. Broekens, J.: Emotion and reinforcement: affective facial expressions facilitate robot learning. In: Huang, T.S., Nijholt, A., Pantic, M., Pentland, A. (eds.) Artifical Intelligence for Human Computing. LNCS (LNAI), vol. 4451, pp. 113–132. Springer, Heidelberg (2007)
4. Chanel, G., Rebetez, C., Bétrancourt, M., Pun, T.: Boredom, engagement and anxiety as indicators for adaptation to difficulty in games. In: Proceedings of the 12th International Conference on Entertainment and Media in the Ubiquitous Era, pp. 13–17. ACM (2008)
5. Chao, C., Cakmak, M., Thomaz, A.L.: Transparent active learning for robots. In: ACM/IEEE International Conference on Human-Robot Interaction, pp. 317–324. IEEE (2010)
6. Chernova, S., Veloso, M.: Interactive policy learning through confidence-based autonomy. J. Artif. Intell. Res. **34**(1), 1 (2009)

7. Chi, M., VanLehn, K., Litman, D., Jordan, P.: An evaluation of pedagogical tutorial tactics for a natural language tutoring system: a reinforcement learning approach. Int. J. Artif. Intell. Educ. **21**(1–2), 83–113 (2011)

8. Cruz, F., Twiefel, J., Magg, S., Weber, C., Wermter, S.: Interactive reinforcement learning through speech guidance in a domestic scenario. In: International Joint Conference on Neural Networks, pp. 1–8. IEEE (2015)

9. Cuayáhuitl, H., van Otterlo, M., Dethlefs, N., et al.: Machine learning for interactive systems and robots: a brief introduction. In: Proceedings of the 2nd Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Perception, Action and Communication, pp. 19–28. ACM (2013)

10. Gallina, P., Bellotto, N., Di Luca, M.: Progressive co-adaptation in human-machine interaction. In: Informatics in Control, Automation and Robotics. IEEE (2015)

11. Giullian, N., et al.: Detailed requirements for robots in autism therapy. In: Proceedings of SMC 2010, pp. 2595–2602. IEEE (2010)

12. Goodrich, M., Colton, M., Brinton, B., Fujiki, M., Atherton, J., Robinson, L., Ricks, D., Maxfield, M., Acerson, A.: Incorporating a robot into an autism therapy team. IEEE Life Sciences (2012)

13. McCullagh, P., et al.: Assessment of task engagement using brain computer interface technology. In: Workshop Proceedings of the 11th International Conference on Intelligent Environments, vol. 19. IOS Press (2015)

14. Knox, W.B., Stone, P.: Reinforcement learning from simultaneous human and MDP reward. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 475–482 (2012)

15. Libin, A., Cohen-Mansfield, J.: Therapeutic robocat for nursing home residents with dementia: preliminary inquiry. Am. J. Alzheimer's Dis. Dementias **19**(2), 111–116 (2004)

16. Modares, H., Ranatunga, I., Lewis, F.L., Popa, D.O.: Optimized assistive human-robot interaction using reinforcement learning. IEEE Trans. Cybern. **46**, 655–667 (2015)

17. Pietquin, O., Lopes, M.: Machine learning for interactive systems: challenges and future trends. In: WACAI (2014)

18. Raya, R., Rocon, E., Urendes, E., Velasco, M.A., Clemotte, A., Ceres, R.: Assistive robots for physical and cognitive rehabilitation in cerebral palsy. In: Mohammed, S., Moreno, J.C., Kong, K., Amirat, Y. (eds.) Intelligent Assistive Robots: Recent Advances in Assistive Robotics for Everyday Activities. Springer Tracts in Advanced Robotics, vol. 106, pp. 133–156. Springer, Heidelberg (2015)

19. Rieser, V., Lemon, O.: Reinforcement Learning for Adaptive Dialogue Systems: A Data-driven Methodology for Dialogue Management and Natural Language Generation. Theory and Applications of Natural Language Processing. Springer, Heidelberg (2011)

20. Senft, E., Baxter, P., Kennedy, J., Belpaeme, T.: SPARC: supervised progressively autonomous robot competencies. In: Tapus, A., André, E., Martin, J.-C., Ferland, F., Ammi, M. (eds.) Social Robotics. LNCS, vol. 9388, pp. 603–612. Springer, Heidelberg (2015)

21. Tapus, A.: Improving the quality of life of people with dementia through the use of socially assistive robots. In: Advanced Technologies for Enhanced Quality of Life (AT-EQUAL 2009), pp. 81–86. IEEE (2009)

22. Torrey, L., Taylor, M.: Teaching on a budget: agents advising agents in reinforcement learning. In: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems, pp. 1053–1060. International Foundation for Autonomous Agents and Multiagent Systems (2013)

23. Tsiakas, K.: Facilitating safe adaptation of interactive agents using interactive reinforcement learning. In: Companion Publication of the 21st International Conference on Intelligent User Interfaces, pp. 106–109. ACM (2016)
24. Tsiakas, K., Huber, M., Makedon, F.: A multimodal adaptive session manager for physical rehabilitation exercising. In: Proceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments. ACM (2015)
25. Wada, K., et al.: Robot therapy for elders affected by dementia. IEEE Eng. Med. Biol. Mag. **4**(27), 53–60 (2008)

http://www.springer.com/978-3-319-47436-6