# Adaptive Interaction of Persistent Robots to User Temporal Preferences

Kim Baraka[1] and Manuela Veloso[2]

[1] Robotics Institute, Carnegie Mellon University
[2] Computer Science Department, Carnegie Mellon University

**Abstract.** We look at the problem of enabling a mobile service robot to autonomously adapt to user preferences over repeated interactions in a long-term time frame, where the user provides feedback on every interaction in the form of a rating. We assume that the robot has a discrete and finite set of interaction options from which it has to choose one at every encounter with a given user. We first present three models of users which span the spectrum of possible preference profiles and their dynamics, incorporating aspects such as boredom and taste for change or surprise. Second, given the model to which the user belongs to, we present a learning algorithm which is able to successfully learn the model parameters. We show the applicability of our framework to personalizing light animations on our mobile service robot, CoBot.

## 1 Introduction

An important part of Human-Robot Interaction (HRI) research aims at finding iconic ways for robots to interact with humans, that are both effective and universal, especially when the interaction has a direct functional role (e.g. communicating intent or instructing the user). Human studies can be helpful in the design of this type of interaction, where one aims at finding one way of interacting which works best on average. On the other hand, there exists another type of interaction whose main purpose is to please or adapt rather than to directly perform a functional role (e.g. pertaining to robot appearance, speech wording, sounds etc.). It is this type of interaction on which we will focus throughout this paper. The assumption is that there can be high variability in the way different users desire or expect to perform this type of interaction with a robot. In this case, a social understanding of the interaction is not very valuable since the problem of interaction choice selection is more a matter of adapting to the user's tastes and is hence theoretically arbitrary. There has been general evidence that personalization of robot appearance and behavior can greatly improve user experience [8] in terms of "rapport, cooperation and engagement" [4], hence the need to move away from the human study paradigm towards automated personalization of the interaction. In this work, we are interested in particular in mobile robots which are persistent over time and which interact with different types of users over an extended time frame. We will use expression through lights on our mobile service robot, CoBot (see figure 1), as a motivating example, but this work can be applied to any form of interaction whose main purpose is to please rather than inform - e.g. voice in speech generation, motion or pose of the robot during interaction, facial expression of a humanoid...

We look at the general problem of learning how to best interact with different individuals based on feedback from the latter. More specifically, we make the following assumptions. (1) A robot repeatedly interacts with different humans whose identity is known by the robot. (2) Every time the robot encounters some individual, it chooses one out of a set of fixed possible options to interact with them. (3) The user has a method of providing feedback for the interaction through a score or rating. In practice, social interactions can be much more complex and require much more context to help adaptation, however there is a wide range of interaction types which do not play a direct functional social role but act more as a complement to the main interaction. For illustration purposes, we use interaction with lights on one of our mobile service robots, CoBot3. CoBot3's expressive lights are being used to enhance the interaction with humans. A finite set of predefined light animations can be used for personalized interaction. The robot, being able to accurately navigate and localize itself accurately in our buildings, can identify users (e.g. by their office number or by recognizing an associated digital signature), hence enabling the personalization of light animations while servicing that user. At the end of each interaction, the touch screen interface may be used to rate the interaction (e.g. by the use of a slider).

Long-term user preferences are however from being static or homogeneous, which is not accounted for in traditional recommender systems. Indeed, being exposed to the same type of interaction for a long time might develop boredom or fatigue for some, while others might value it for its predictability. To summarize, general static preferences change from individual to individual, but preference dynamics are equally important in a long-term setting. In this paper, we propose to learn, for a given user, both sets of preferred interaction options and time-related quantities which would dictate the preferred dynamics of interaction.

The paper is divided into three main parts. In the first part, we introduce three user models which capture different possible profiles in relation to the appreciation of "change" in the way the robot interacts with the user. These are formulated in terms of evolution of the reward from the user as a function of the possible sequences of interactions options used when interacting with that user. In the second part, we present our algorithms to learn the model parameters, assuming we are given the user model. In the third part, we show the applicability of this work to our mobile service robot, CoBot, which is deployed in our buildings and uses expressive lights for improved interaction with humans.

## 2 Related work

Apart from simple customization during usage [8], recent work has looked at autonomous personalization of HRI based on previous interactions with the same user. Examples include a snack delivering robot which uses data from past interactions to personalize the future interactions [4] or a humanoid robot learning different models for expressing emotion through motion, which it is then able to use for personalizing expression of emotions [9]. Furthermore, the idea of self-initiative in a robot has been explored by learning ways of acting in the world depending on user verbal feedback on the current "state of the world" [7]. Finally, user modeling for long-term HRI, a focus of the current paper, has been looked at using archetypes of real users called *personas*, which encode traits of potential users in terms of interaction preferences [3]. Some authors also looked at ways of learning long-term behavior by identifying social primitives that are important when the novelty aspect of interaction vanishes [12] or matching personalities between robot and user [11]. However, these works focus more on the social aspect of the interaction rather on the intelligence of the adaptation from a generic point of view, making their applicability and generalization poor in different types or modes of interaction. In this work, we would like to decouple the nature of the interaction options with the generic adaptation mechanism, which can then be tuned based on the nature of the interaction and the user's response to it.

In the problem we consider, we will assume that user data are limited to rewards at every interaction (in the form of a rating submitted by the user), making it comparable to a recommender system learning user preferences and suggesting new items [1]. However, the algorithms used in such systems do not take into account the dynamics of preferences (boredom, habituation, desire for change etc.). In the field of automatic music playlist generation, the concepts of diversity and serendipity have been mentioned [2]. However, no viable solution has yet been proposed to address this problem. Also, the idea of exploration in music recommender systems has been studied [10], but it does not apply to our problem since we assume the number of interaction options to be relatively small. In a robotics application, the need for adaptive interaction that takes into account habituation has been recently formulated for empathic behavior [12] (in this paper, we take a more general approach). Going back to the problem of preference dynamics, our problem can formally be compared to the restless multi-armed bandit problem where rewards are non-stationary and which is generally known to be P-SPACE hard [5]. In this work, we restrict the rewards to evolve according to one of three models, which makes the problem of learning the model parameters easier to solve.

## 3 Formalism and user modeling

In this section, we start by presenting the formulation of the problem at hand and move on to introduce three models of dynamic preferences corresponding to three different possible profiles of users.

### 3.1 Problem setting

Time is discretized into steps $i = 1, 2, 3, ...$, where each time step represents an encounter between the robot and the user. We assume that the encounters are of an identical nature or serve the same functional role (for example the robot is always delivering an object to the user's office). Also, for simplicity, we do not worry about the actual time interval between two consecutive steps (these could be for example different days or different times within the same day). At every time step, we assume the robot has to choose one out of a set of $n$ possible actions corresponding to interaction options. In the context of light animations, the different actions represent different animations in terms of speed, color or animation patterns. Let $\mathbf{A} = \{a_1, ..., a_n\}$ represent the set of possible actions. After every encounter, we assume that the user provides a rating $r^{(i)}$ to the interaction where $r^{(i)} \in [0; 10]$. The reward is assumed to be corrupted by additive white Gaussian noise: $r = \bar{r} + \epsilon$ where $\epsilon \sim N(0, \sigma^2)$. The noise can come from the following sources: (1) inaccurate reporting of the user's true valuation, (2) mistakes when using the user interface (e.g. slider) to report the reward and (3) failure to remember previous ratings resulting in inconsistent ratings.

Our goal is to learn, for a specific user (with a specific reward model), which action to take next given the history of actions and rewards. The problem can hence be compared to the Multi-Armed Bandit problem where a single player, choosing at each time step one to play one out of several possible arms and gets a reward for it, aims to maximize total reward (or equivalently minimize total regret) [5]. In our case, the rewards are stochastic and non-stationary and the arms or actions, corresponding to the different interaction options, are relatively few. From now on, we will use "actions" and "interaction options" interchangeably.

### 3.2 Modeling dynamic user preferences over time

We now introduce three user models which we think span well enough the spectrum of possible profiles, inspired by variations along the "openness" dimension of the five-factor model in psychology [13]. These models we crafted take into account both properties of preferred actions sets and time-related quantities dictating the evolution of rewards depending on the action sequences. Figure 1 shows sample ideal sequences for lights on our robot, CoBot3, for each of the three models on different days in which the robot visits a person's office to deliver coffee. For the three models presented below, we use $\mathbf{A}_{\text{pref}}$ to denote the set of preferred actions (regardless of the sequence of actions in which they fall).

**Model 1: The "conservative"** This type of user wants to stick to one option denoted by $a^*$, but appreciates surprises from time to time at some frequency. A surprise means taking for one time step an action $a \neq a^*$ in a set of preferred "surprise actions" $A_{\text{surp}} \subset \mathbf{A}$. When $a^*$ is repetitively shown in a sequence (we call sequences of the same action homogeneous sequences), the reward $\bar{r}$ starts out as a constant ($r_{\text{max}}$) and after $T$ time steps starts decreasing, due to boredom, with a linear decay rate $\alpha$ until it reaches $r_{\text{min}}$, after which it remains constant. For homogeneous sequences of the non-preferred actions (i.e. actions in $A \setminus \{a^*\}$), the reward starts at a value $r_{\text{non-pref}}$ and decreases exponentially to zero with time (indicating that the user very quickly gets bored) with some decay rate $\beta$. In summary, the model parameters are:

- $a^*$: the action with the maximum value of $E[\bar{r}]$. A homogeneous sequence of $a^*$ actions is referred to from now on as a **p-sequence**.
- $\mathbf{A}_{\text{pref}} = \{a^*\}$
- $\mathbf{A}_{\text{surp}}$: set of actions suitable for surprises, defined as $\{a : E[\bar{r}_a] > r_{\text{th}}\}$, where $r_{\text{th}}$ is a threshold value.
- $T$: optimal length of the homogeneous sequence of the preferred action, after which the user starts getting bored. If the robot always alternates between p-sequences and surprises, $T$ can also be seen as a between two consecutive surprises. T is assumed to be a random variable uniformly drawn in a window $[T_{\min}, T_{\max}]$ every time a new p-sequence is started.
- $\alpha$: linear reward decay rate in a p-sequence whose length exceeds $T$.
- $r_{\max}$: constant reward for p-sequences of length less than or equal to T.
- $r_{\min}$: lower clipping value for reward in p-sequences. A good value for is 5, which means that the user is neither rewarding nor punishing the robot for taking their preferred action for too long.
- $r_{\text{non-pref}}$: initial reward value when starting a homogeneous sequence that is not a p-sequence. If the previous homogeneous sequence is a p-sequence, $r_{\text{non-pref}}$ is a function of the length of the p-sequence $l$ as follows: if $l \geq T_{min}$ we assume that the user is expecting a surprise which will provide some maximal reward $r_{\text{non-pref,max}}$. When $l < T_{min}$, we expect the surprise to be disliked, so we decrease the surprise reward linearly: $r_{\text{non-pref}} = r_{\text{non-pref,max}} . (1 - \frac{T_{min}-l+1}{T_{min}})$. If the previous homogeneous sequence is not a p-sequence, $r_{\text{non-pref}}$ is a constant $r_{\text{non-pref,base}}$.
- $\beta$: exponential reward decay rate for a homogeneous sequence that is not a p-sequence.

**Model 2: The "consistent but fatigable"** This type of user values consistency in actions taken but needs shifts from time to time. It is the profile where there always needs to be an uninterrupted routine but this routine has to be changed after some time. The user has a set of preferred actions which he expects to see in long sequences. These sequences alternate between the different preferred options after some time spent sticking with one of the options. We assume the same model of boredom used in the previous section, namely the reward starts decaying linearly for the preferred actions after some time interval $T$. There is no surprise factor associated with this model since we assume that the user does not appreciate surprises.

The parameters of this model are the following (no description provided means the parameters are the same as in the "conservative" model):

- $\mathbf{A}_{\text{pref}} = \{a_1^*, ..., a_m^*\}$, where $m \geq 2$. p-sequences in this model are defined to be homogeneous sequences formed using one action in $\mathbf{A}_{\text{pref}}$.
- $T$: optimal length of a p-sequence, after which the user starts getting bored. T is assumed to be a random variable uniformly drawn in a window $[T_{\min}, T_{\max}]$ every time a new p-sequence is started.
- $\alpha$, $r_{\max}$ and $r_{\min}$: idem

– $r_{\text{non-pref}}$: initial reward value when starting a homogeneous sequence that is not a p-sequence. A constant in this model.
– $\beta$: decay rate of the reward for a homogeneous sequence that is not a p-sequence.

**Model 3: The "erratic"** This type of user is mainly interested in change, in both action selection and time-related parameters. They have no clear preferences over the possible options but require the actions to change according to some average rate not restricted to a window as in model 1 and 2. We assume that at every step the user has some fixed probability $p_{\text{sw}}$ to desire a switch to a different action, independently of anything else. Hence the optimal length $T$ of homogeneous sequences follows the distribution: $p(T = t) = (1 - p_{\text{sw}})^{t-1} p_{\text{sw}}$ (for $t \geq 1$), whose average $\mu_T = 1/p_{\text{sw}}$, making $\mu_T$ a sufficient statistic. Similar to previously, the reward decreases linearly after T time steps in a homogeneous sequence.

## 4 Learning model parameters from user feedback

Now that we have presented the three user models that we consider, we look at the problem of learning their parameters from user reward sequences. Once these parameters become known, we can then generate personalized sequences of actions maximizing cumulative reward for a specific user. In what follows, we assume that the model to which a particular user belongs to is known a priori. In practice, this can be achieved by prompting the user to select one profile which described them best, or through a set of questions similar to a personality test.

Note that although we have previously raised the problem of dealing with the non-Markovian aspect of user preferences (meaning that the reward of a certain action depends on the history of previous actions), in the way we have modeled the user profiles in the previous section, the model parameters encode the preference dynamics. These parameters are assumed to be unchanged as time goes by, hence we have effectively turned the dynamic problem into a Markovian one. Next, we describe the learning procedure for each of the user profiles introduced.

### 4.1 Profile "conservative"

In order to learn the parameters of this model, we divide the learning process into two phases: one phase for learning preference sets and the other for learning the time-related parameters. The parameters the agent performing the actions needs to learn are: $a^*$, $\mathbf{A}_{\text{surp}}$, $T_{\min}$ and $T_{\max}$.

**Phase 1: Learning preference sets** In this preliminary phase, actions are uniformly drawn from $\mathbf{A}$ until each action is taken at least $n_{\text{th}}$ times, where $n_{\text{th}}$ depends on the noise variance estimate $\tilde{\sigma}^2$ and on our target confidence value (for all practical purposes, we use $n_{\text{th}} = 10$). Note that randomization of the sequence of actions to be taken is crucial in this phase since the previous actions


Fig. 1: Sample preferred of animation sequences for the user models presented

can have an influence on the reward of the current action and we would like to dilute this effect. Once we have an average reward estimate for each action, we select $a^*$ to be the action with the maximum estimated reward and $\mathbf{A}_{\text{surp}}$ to be the set of all actions whose reward estimates exceed the set threshold $r_{\text{th}}$, where the value of $r_{\text{th}}$ has to ensure that $|\mathbf{A}_{\text{surp}}| \geq 1$. It assumed that the set of best actions to be used for surprises will score high in this phase as well.

**Phase 2: Learning time-related parameters** In order to learn the two parameters of interest $T_{\min}$ and $T_{\max}$, the agent first learn estimate the mean and variance of $T$ ($\mu_T$ and $\sigma_T$ respectively) and uses them to infer the parameter values. To achieve this, the agent follows p-sequences until a need for surprise is detected (more details below). A surprise is restricted to taking an action in $\mathbf{A}_{\text{pref}}$ for one time step following a p-sequence. After a surprise, the agent reverts back to following a p-sequence until another surprise is decided upon.

The learning procedure goes as follows: when in a p-sequence of actions, if a downward trend in reward is detected, show a surprise chosen uniformly from $\mathbf{A}_{\text{pref}}$. Since the reward is noisy, a smoother is needed to filter out high frequency noise in the data. We use an exponentially weighted moving average (EWMA) [6] with fixed sample size $s$, combined with a threshold detector, to detect a downward trend in the reward of the p-sequence. The threshold used in the threshold detector depends on the estimated noise variance in the reward $\tilde{\sigma}^2$. Every time a downward trend is detected, we record the estimated $T$ value associated with the p-sequence. Once enough surprises are chosen, we would have accurate enough estimates of $\mu_T$ and $\sigma_T$, which can be used to find the time-related parameters as follows: $\tilde{T}_{\min,\max} = \tilde{\mu_T} \mp \frac{\sqrt{12\tilde{\sigma_T}^2+1}-1}{2}$.

Note that there is a lag associated with the moving average trend detector. This lag is equal to half the $\lfloor \frac{s}{2} \rfloor$ and $\tilde{\mu}_T$ needs to be adjusted to account for it. Also, for a small number of data points, we might be overestimating $\sigma_T$. Hence we set $\tilde{\sigma_T}$ to be half the estimate of the standard deviation in the values of $T$. This way we impose a more conservative restriction on the values of $T$ which will ensure that $[\tilde{T}_{\min}, \tilde{T}_{\max}] \subset [T_{\min}, T_{\max}]$.

### 4.2 Profile "consistent but fatigable"

Similar to that of the previous model, the learning procedure is still separated into the two phases. However, as far as action selection is concerned, since there is no surprise but only a change factor in this model, the termination of a p-sequence of $a_i$ consists in starting a new p-sequence with an action chosen uniformly in $\mathbf{A}_{\text{pref}} \setminus \{a_i\}$. The first phase for learning preference sets uses the same procedure as before, except that once the average reward estimates are obtained, we set $\mathbf{A}_{\text{pref}}$ to be the set of animations with a reward estimate above $r_{\text{th}}$ (possibly different than the one used in the "conservative" model). Here again, the threshold value should be set such that the cardinality $m$ of $\mathbf{A}_{\text{pref}}$ is at least 2. The second phase for learning time-related parameters is similar to the one used in the previous model.

### 4.3 Profile "erratic"

For this type of profile, no sets of preferred actions need to be learned since we assume that the user has no clear preferences between the different actions. Hence, the only parameter to learn is the probability of switching $p_{\text{sw}}$. The action selection algorithm is identical to the "consistent but fatigable" model, with $\mathbf{A}_{\text{pref}} = \mathbf{A}$. $\mu_T$ can also be estimate as before, and once a good estimate is obtained, we infer our parameter $p_{\text{sw}}$ as follows: $\tilde{p}_{\text{sw}} = \frac{1}{\tilde{\mu}_T}$.

### 4.4 Action sequences generation

The learning phase stops when the parameters are learned with some target confidence value. The latter comes in our case mainly from the error rate of the EWMA and depends on the various parameters including noise variance. Once the parameters are learned, appropriate stochastic sequences can be generated according to the estimated parameter values. For models "conservative" and "consistent but fatigable", we uniformly draw a value of $T$ in the estimated window. For model "erratic", we follow the same action with probability $1 - p_{\text{sw}}$ and uniformly switch to another action with probability $p_{\text{sw}}$. In this exploitation phase, the feedback requirements can be reduced or eliminated, since we have all the parameters needed to generate optimal sequences which will maximize the cumulative reward for the given user. In practice, occasional user feedback (e.g. asking for a reward) can be used to confirm the model and parameters. We will not provide more detail about this exploitation phase since the focus of this work is on the learning aspect. However, notice that in the learning phase we are already generating sequences which are not too far from optimal.

## 5 Results

In this section, we present a few results showing our implementation of our simulated user's preference dynamics and our algorithm's ability to learn the different model parameters. Figure 2 shows the evolution of the learning process for single instances of the three models. We consider 8 possible actions arbitrarily labeled 1 through 8. Phase 1 of the learning algorithm can be clearly distinguished in the first two models, after which the algorithm learns the set $\mathbf{A}_{\text{pref}}$ ($\{a_4\}$ for model "conservative" and $\{a_2, a_4, a_6\}$ for model "consistent but fatigable"). Once it identifies the preferred sets, the algorithm is also able to adapt to the preferred action dynamics. Notice that whenever there is a notable decrease in the reward, a change is performed, whether creating a temporary "surprise" (a), changing to another steady option (b) or creating erratic change (c).

The simulation was run over 350 time steps with the following parameter values for illustrative purposes. $T_{\min} = 20$ and $T_{\max} = 30$ for the first two models and $p_{\text{sw}} = 0.8$ for the third model. The noise variance $\sigma^2$ was set to 0.05. Here are a few results over 1,000 randomized trials.

**-Model "conservative":** % error in $\tilde{\mu_T}$: 3.5% ; $\tilde{T}_{\min} = 20.94$; $\tilde{T}_{\max} = 30.81$. After rounding, the estimated interval is contained in the true interval.

**-Model "consistent but fatigable":** % error in $\tilde{\mu_T}$ : 2.67% ; $\tilde{T}_{\min} = 21.72$; $\tilde{T}_{\max} = 26.95$. The rounded estimated interval is contained in the true interval.

**-Model "erratic":** $\tilde{\mu_T} = 12.67$, therefore $p_{\text{sw}} = 0.079$ (0.13% error).

The algorithm was able to learn all the parameters with reasonable accuracy.

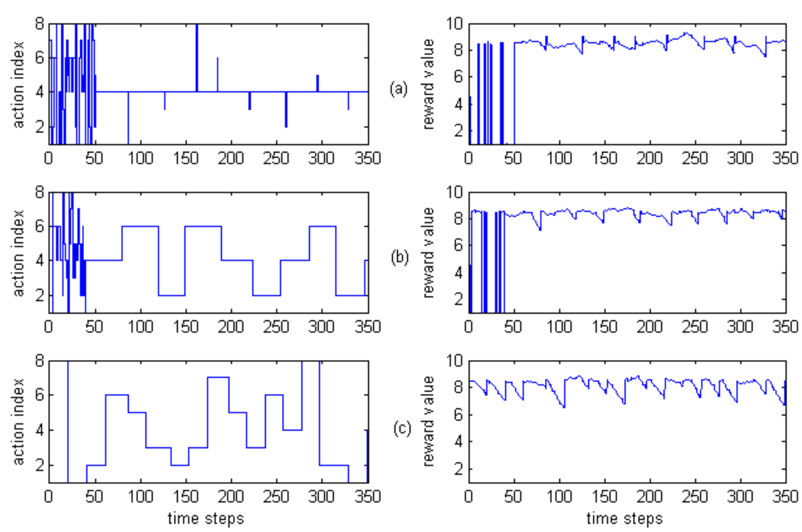


Fig. 2: Simulation results showing sequences of actions taken by the agent and the corresponding reward sequences from a simulated user belonging to: (a) model "conservative", (b) model "consistent but fatigable" and (c) model "erratic"

## 6 Conclusion and future work

We have presented three models for dynamic long-term user preferences, which capture aspects of boredom and appreciation for change or surprise. Given which model a specific user belongs to, our algorithm enables the robot to learn the model parameters using the sequence of rewards given by the user. Our results show that the agent is able to learn the parameters of the model reasonably well and in a relatively short number of time steps for all three models. Our algorithm is robust to noise, but further experiments are needed to evaluate the degradation in performance as the noise increases. In the future, we plan to enable the robot to also learn which model the user belongs to from the reward sequences themselves. Also, allowing a mixture of the three models with weights to be learned (although it is not exactly clear whether it is a viable idea in a social setting) could diversify the space of sequences generated and alleviate the problem of forcing the users to categorize themselves. Furthermore, requesting a reward from the user at each encounter might have a negative social impact; to overcome this, we could investigate efficient sampling methods to gather rewards in a sparse manner while maintaining accuracy in the learning process.

## References


1. J. Bobadilla, , F. Ortega, A. Hernando, A. Gutierrez, "Recommender systems survey", Knowledge-Based Systems, Vol. 46, 109-132. 2013.
2. G. Bonnin, D. Jannach, "Automated generation of music playlists: Survey and experiments", ACM Computing Surveys (CSUR), Vol. 47 Issue 2, 26. 2014.
3. J. Campos, A. Paiva, "A personal approach: the *persona* technique in a companion's design lifecycle", INTERACT 2011, Lecture Notes in CS Vol. 6948, 2011.
4. M. Lee, J. Forlizzi, S. Kiesler, P. Rybski, J. Antanitis, S. Savetsila, "Personalization in HRI: A Longitudinal Field Experiment", HRI, 2012.
5. H. Liu, K. Liu, Q. Zhao, "Learning in A Changing World: Restless Multi-Armed Bandit with Unknown Dynamics", ARXIV, 2010.
6. J. Lucas, M. James, M. Saccucci. "Exponentially weighted moving average control schemes: properties and enhancements." Technometrics 32.1, 1990.
7. M. Mason, M. Lopes, "Robot Self-Initiative and Personalization by Learning Through Repeated Interactions", HRI, 2011.
8. J. Sung, R. Grinter, H. Christensen, "Pimp My Roomba: Designing for Personalization", CHI 2009.
9. M. Vircikova, M. Pala, P. Smolar, P. Sincak, "Neural Approach for Personalised Emotional Model in Human-Robot Interaction", WCCI 2012.
10. X. Wang Y. Wang, D. Hsu, Y. Wang, "Exploration in Interactive Personalized Music Recommendation:A Reinforcement Learning Approach", Mult. Comput. Commun. Appl., Vol. 11, 7. 2014.
11. A. Tapus, C. Tapus, and M. Mataric. "User-robot personality matching and assistive robot behavior adaptation for post-stroke rehabilitation therapy." Intelligent Service Robotics 1.2 (2008): 169-183.
12. Leite, Iolanda. "Long-term interactions with empathic social robots." AI Matters 5.1 (2015): 13-15.
13. Goldberg, Lewis R. "An alternative" description of personality": the big-five factor structure." Journal of personality and social psychology 59.6 (1990): 1216.