

# Learning Robot Policies Using a High-Level Abstraction Persona-Behaviour Simulator

Antonio Andriella<sup>1</sup> Carme Torras and Guillem Alenyà

**Abstract**—Collecting data in Human-Robot Interaction for training learning agents might be a hard task to accomplish. This is especially true when the target users are older adults with dementia since this usually requires hours of interactions and puts quite a lot of workload on the user.

This paper addresses the problem of importing the Personas technique from HRI to create fictional patients' profiles. We propose a Persona-Behaviour Simulator tool that provides, with high-level abstraction, user's actions during an HRI task, and we apply it to cognitive training exercises for older adults with dementia. It consists of a Persona Definition that characterizes a patient along four dimensions and a Task Engine that provides information regarding the task complexity. We build a simulated environment where the high-level user's actions are provided by the simulator and the robot initial policy is learned using a Q-learning algorithm. The results show that the current simulator provides a reasonable initial policy for a defined Persona profile. Moreover, the learned robot assistance has proved to be robust to potential changes in the user's behaviour. In this way, we can speed up the fine-tuning of the rough policy during the real interactions to tailor the assistance to the given user. We believe the presented approach can be easily extended to account for other types of HRI tasks; for example, when input data is required to train a learning algorithm, but data collection is very expensive or unfeasible. We advocate that simulation is a convenient tool in these cases.

## I. INTRODUCTION

According to the Alzheimer's report of 2018 there are 52M people worldwide living with dementia [1]. Since the incidence of dementia rises with age, the expected growth in the worldwide older adults population in the next decades has been projected to reach about 152 million people by 2050. The current cost of this disease is about 1B dollars per year, and it is going to double by 2030. For these reasons, eldercare is rapidly becoming one of the most daunting healthcare challenges of our time.

The lack of an effective pharmacological therapy has moved the focus of many researchers toward different non-pharmacological treatments. Cognitive Therapy (CT) is one of such intervention. According to [2], there is evidence from clinical trials which indicates that CT may be effective in lowering dementia risk and slowing the rate of decline.

Socially Assistive Robotics (SAR) aims to endow robots with the ability to help people through individual social assistance, rather than physical, in convalescence, rehabilitation,



Fig. 1: Tiago robot interacting with a user while he is playing a cognitive exercise. The initial policy of the robot has been generated learning the user's profile using the Persona-Behaviour Simulator.

training, and education [3]. In order to be effective, every kind of therapy needs to be tailored according to the user needs, this being the reason why SAR can help to bridge the gap when human assistance is not available.

One of the main challenges in SAR, and in general in Human-Robot Interaction (HRI), is the difficulty in collecting user data such as configuring the robot in order to provide the most suitable assistance related to the given user. This essential requirement creates the need for long experimental studies aimed to collect as much data as possible. This method is known as data-driven [4]. Classical approaches includes complex state machines and heavily manually hand-crafted planning systems which combine speech and gestures necessary to display a behaviour are not a feasible solution, considering the amount of possible states a robot might be in. Other techniques such as Learning by Demonstration (LbD) [5] and Reinforcement Learning (RL) [6] algorithms allow the robot to learn their behaviour without the need to explicitly preprogram them. However, also these techniques require a considerable amount of interactions that still don't solve our original problem: how can we reduce the user's burden and teach the robot the right behaviour?

In order to overcome this limitation, we propose the usage of a knowledge-driven approach that does not need an involvement of real patients for the creation of a data collection [7], [8], [9]. We make use of the concept of

\*This work has been partially funded by the EU project SOCRATES H2020-MSCA-ITN-721619, by the Spanish Ministry of Science and Innovation HuMoUR TIN2017-90086-R, and by the State Research Agency through the Mar de Maeztu Seal of Excellence to IRI (MDM-2016-0656).

<sup>1</sup>A. Andriella, C. Torras, G. Alenyà are with Institut de Robòtica i Informàtica Industrial, CISC-UPC, C/Llorens i Artigas 4-6, 08028 Barcelona, Spain. {aandriella, torras, galenya}@iri.upc.edu

Personas [10] for modelling human users. Personas can be seen as a tool for creating fictitious user representations in order to embody different behaviours. Then, we create a Persona-Behaviour Simulator (PBS) and we define a Social Assistive Robotic Agent (SARA) that can generate multiple interactions and learn from them.

In this paper, we build on our previous work [11], where a robot is endowed with the abilities to assist dementia patients during cognitive exercises (see Figure 1). In [11], we define two loops of interaction: the first one in which the caregiver sets the physical and mental user impairment and the initial robot behaviour; the second one in which the robot interacts with a patient using a hand-crafted policy and providing him with adaptive levels of assistance based on his performance, while he is playing a cognitive exercise. In this article, we extend this system by: i) providing an easy tool for the caregiver to set up the patient's profile through the concept of Persona (first loop of interaction) and ii) learning the robot initial policy (second loop of interaction).

The main contributions of this paper are:

- proposal of a PBS with a high-level of abstraction for user's actions that can be employed in most HRI tasks
- validation of a PBS for learning a SARA initial policy in a cognitive training scenario.

The main idea underlying the presented work is that, through a simulated framework, we learn the robot behaviour for a given user that matches his generated Persona profile. Patient involvement occurs only in the fine-tuning stage so we can optimize the robot policy on the individual's cognitive needs and preferences. This is very relevant when the people involved in the experiment are older adults with cognitive impairment. Asking them to be involved several times or even days in the same task can potentially create situations of distress and pressure. Additionally, it is not always a good strategy to leave patients interacting with a robot that lacking proper training, since that might affect the patients' future judgement on the robot and their engagement with it.

The presented approach is not intended to be exhaustive, since creating a cognitive model of the user is a complex task. Nonetheless, the aim of this paper is to present an alternative approach to the classic ones and deserves to be explored. Finally, the proposed method has the advantage of evaluating from the simulation the effectiveness of the algorithm, its parameters sensitivity, and the quality of the assistance provided.

## II. RELATED WORK

The concept of Personas has been widely used in Human-Computer Interaction (HCI) while very few works have investigated the possibility to use this concept in HRI and to best of our knowledge no works have explored its use for learning a robot initial policy.

### A. Personas in HRI

The concept of Personas has been defined in psychology by Jung [12], where the word persona is meant as the different social behaviours (social masks) that an individual

can wear among various situations. The concept of Persona has been then used by Alan Cooper [10] in HCI. This technique focuses on the idea of understanding the users needs and objectives. According to Cooper definition, a Persona is a fictional, detailed user model that represents archetypical users. The special aspect of a Persona is that his description is not intended to be fully descriptive, but it uses the area of focus or the task domain as a lens to highlight the relevant attitudes and the specific context associated.

One of the most relevant works using Persona in HRI in the context of activities of daily living is presented by Dunque *et al.* [13]. In their work, they define a Persona-Based Computational Behaviour Model for developing SAR in living environment. They define which variables describe a Persona and which robots features should be defined to adapt to a given Persona. Dos Santos *et al.* [14] describe a methodological approach for creating Personas in designing new features for robots. To this end, they conduct experiments to collect data from questionnaires, and video analysis from users while interacting with the robot. Shulz *et al.* [15] outline techniques to get more information about assistive technology and to include people with disabilities in the Persona creation process. In particular, with the aim to have a universal design for people with disabilities they define four main features for a Personas: vision, hearing, movement and cognitive impairments.

In this article, we define our Persona in the context of a cognitive training exercise based on Duque *et al.* [13] proposal and according to Shulz's definition of Persona with disabilities [15], we define a Persona with dementia along four dimensions that are: memory, attention span, reactivity and hearing. Differently from Duque *et al.* [13], we go a step further providing a real implementation of the behavioural Persona with a high-level of abstraction on the user's actions. Moreover, we perform experiments to validate the feasibility of our proposed approach.

### B. RL for learning robot adaptive behaviour

RL algorithms have been widely used in SAR to learn a user-specific policy in order to improve the effectiveness of the interaction and thus the assistance provided by the robot. Hemminghaus *et al.* [16], present an approach to generate social behaviour in a robot (Furhat) in an adaptive way while it is interacting with a user in a memory game. Their main objective is to evaluate if the employed social robot Furhat is able to learn which interaction modalities among gaze, facial expression, head gesture and speech are more suited for a given user. They propose a Q-learning algorithm in which the reward is defined based on the success of the user action and the discounted from the assistance received. Leite *et al.* [17], [18] propose a Multi-Armed Bandit (MAB) algorithm able to provide robots empathic responses to particular preferences of a child who is interacting with the robot during a chess game in order to keep him engaged over time. Gao *et al.* [19], extend the work of [18], proposing a robotic system that is able to learn using Exp3, a MAB algorithm, the most effective levels of assistance that maximize the

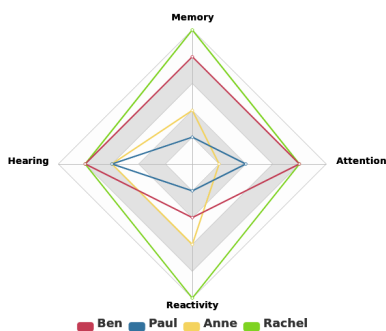


Fig. 2: Example of different Personas profiles according to our defined dimensions.

users performance during the task in an educational scenario. Gordon *et al.* [20] develop a framework in which a social robot provides assistance to children play a second languages learning game or tablet. The proposed state-space formulation includes valence, arousal and engagement detected from the child. These values are also used as a reward.

In our scenario, we use Q-learning with a different state space and action definition. For example, we don't evaluate the interaction modalities [16] but instead, we propose increasing levels of assistance. Moreover, our formulation of reward takes into account not only the assistance provided by the robot but also the complexity of the task and the number of attempts of the user in a given stage of the exercise. Differently, from [18] and [20] we are not using valence and arousal to guide the learning process while as [19] we take into account users performance as well as the assistance received by the robot [16].

All the presented approaches require quite a large amount of interactions to converge to the optimal policy. For instance, in [16] the policy doesn't fully converge and Gordon *et al.* [20] need more than 10,000 of iterations to learn the optimal policy. Since SAR are designed to interact with vulnerable populations, it is not feasible to train a robot using a high number of interactions with users. To minimize the number of iterations required for behaviour learning, we propose to use a PBS in combination with a SARA that uses Q-learning algorithm to quickly learn an initial policy for selecting assistive robot behaviours to display based on the user's profile.

### III. PERSONA DEFINITION

In this Section, we present our definition of Persona. The idea behind the concept of Persona is two-fold. On the one hand, there is an attempt to overcome the problems related to data collection, while on the other hand, we aim to provide each caregiver with an easy way to setup the initial robot behaviour.

The dimensions along with we model the Persona are (see Figure 2):

- memory:** the patient's cognitive impairment
- reactivity:** the patient's physical reactivity
- attention span:** the patient's ability to keep focus

**hearing:** the patient's capability to hear suggestions

The four features have been defined in collaboration with doctors and caregivers specialized in treatment of patient with dementia and are based on [15]. Each feature is defined on a scale of 1 to 5, where 1 means the patient does not have that ability and 5 means the patient has full capability for that skill. Our PD could be enlarged with additional characteristics if needed, such as:

**personality:** the patient personality trait; the caregiver can choose between introverted and extroverted.

**safety\_risk:** the safety behaviour of the patient. The caregiver can set it to low, medium and high. A low value means the caregiver believes the patient will not put himself in danger during the interactions with the robot. A high value means the robot needs to be careful while performing movements and be ready to react to user's unsafe behaviours.

### IV. TASK ENGINE

The Task Engine (TE) is the module that manages the information related to the task itself. There are 2 distribution functions that characterize the TE:

**complexity:** it is defined as the probability to guess the right move at a given state  $s$  of the game. Depending on the task complexity, we model that probability according to one of these distributions: normal, binomial, gamma and Poisson. Each Persona has a different parameters initialization for a given distribution, namely, given a state  $s$  different Personas will have different task complexity probability in  $s$ .

**attempt:** it is defined as the probability to guess the right move after  $n$  attempts on the same token. As for the case of complexity, the attempt function also takes the form of one of the previous distributions. The main difference is that attempt will be reset after each correct move of the Persona.

The current TE is thought to be as generic as possible in order to be used in scenarios unlike the ones discussing here. A different task, for example, can be a physical exercise: the TE formalizes the complexity as the sequence of movements the user has to perform in order to complete it. Preparing meals at home or managing a person's household are example of Instrumental Activities of Daily Living that can be simulated using the presented TE. The therapist needs to define the task complexity over the different steps to complete the task. For example, in the case of the task preparing meal, the TE complexity will be the number of ingredients necessary to prepare the food. Initially it would be harder considering the number of ingredients available. However, choosing the right one with the robot assistance, will decrease the complexity progressively.

### V. PERSONA-BEHAVIOUR SIMULATOR

The PBS is responsible to generate with high-level of abstraction user's actions given a PD setting (see Section III) a TE (see Section IV) and the level of assistance provided by

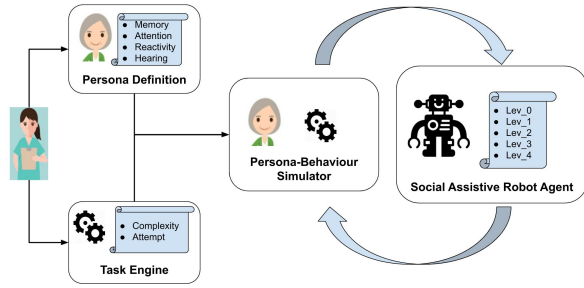


Fig. 3: Main components of the Framework.

a SARA. In Figure 3, we show the framework. The caregiver sets the initial Persona skills according to the patient profile (PD). Then he sets the task parameters based on the kind of task the patient needs to accomplish (TE). During each episode, the SARA generates multiple interactions tailored to assist the patient in solving the task. The output of the simulator can be one of following high-level abstraction Persona’s actions:

$[move\_outcome; reaction\_time]$ , where  $move\_outcome \in \{fright\_move; wrong\_move\}$ . It is computed taking into account memory and reactivity Persona’s dimensions.

$[no\_action; timeout]$ . It is computed taking into account attention span value Persona’s dimension.

It is worth to mention that the high-level of abstraction used to define user’s outcomes will provide a generic tool that is not limiting the current simulator to a particular HRI context (as mentioned in Section IV).

The core of the PBS is the computation of the user probability to perform a given action. The system is composed by a static and a dynamic component. The PD represents the static part of the simulator since we assume these features will not change during the task. It should be noted that in the current implementation, only memory, reactivity and attention dimensions are taken into account for generating a Person’s high-level action. On the contrary, the TE represents the dynamic part of the simulator, since it takes into account the variation of the task complexity over time and it affects the estimation of the user probability to guess the right action in a given state  $s$ .

Another additional aspect worth mentioning is that our simulator can be used also within one-shot tasks such as task in which the episode length is 1. An example of this is the robot remembering the user medication assumption routine. In this task, the high-level abstraction PBS might be configured to return if the user took or not a medication. Since the user’s actions (took or not the medication) can be affected by other external variables, our simulator is still a valid solution for generating data. For instance, let’s consider a user at a specific time of the day, sitting in front of the television when he should take the medication. What would be the outcome then? Since our simulator is task-independent, if we can formalize the task logic in the TE, then the simulator will be able to generate different user’s

outcomes based on the defined external conditions.

#### A. Relationship Persona’s Features with Robot’s behaviour

It is important to define a relation between a PD and a SARA. In other words, we need to provide the caregiver with hints on the effects for each Persona feature is set on the robot’s behaviour. Note that features defined in Section III are generic, and potentially extendable to other tasks.

For each Persona feature, we list below a relationship associated with the robot characteristics:

Persona **memory and reactivity** ! Robot **levels of assistance**: the worse the mental condition of the patient the more the robot will provide assistance.

Persona **attention span** ! Robot **re-engagement actions**: if the patient loses attention, the robot can try among several strategies to engage with the patient again.

Persona **hearing** ! Robot **gestures/speech balancing**: the lower the value, the more the robot will assist the patient preferring gestures rather than speech.

Persona **personality trait** ! Robot **personality behaviour**: The robot can behave in a way that is more suited for the patient personality. Consequently, the match between the user personality and the robot behaviour is defined according to the literature [21], [22].

Persona **safety risk** ! Robot **safety levels**: The higher the level, the more the robot needs to be safe while interacting with the user.

## VI. COGNITIVE TRAINING USE-CASE

As already mentioned in Section IV, our idea is to develop a framework that is as much generic as possible. However, for the sake of clarity, it is better at this point to introduce our particular use-case to provide intuitive examples.

#### A. SKT: the Cognitive Training Task

Our use-case [11] is to develop a robotic system easy to configure from the caregiver (first loop of interaction). The robot can administer cognitive exercises based on the Syndrom-Kurztest (SKT), encouraging and motivating the user through speech and gestures (second loop of interaction). The SKT is a cognitive test to evaluate patients attention and memory. Based on it and in agreement with our partner hospital, we develop a series of cognitive training exercises. One of them is called *sorting blocks*. The objective of this exercise is to sort tokens in ascending/descending order on a board making as few mistakes as possible while minimising the intervention of the robot. Every time the user commits an error, the robot moves the token back to its initial location and provides some assistance.

#### B. SARA adaptative behaviour

The objective of our simulation is to learn an assistive policy for a given patient profile by optimizing the levels of assistance in order to complete the test. The levels of assistance provided by the robot are:

**LEV\_0**: the robot alerts the user that his turn has began

**LEV\_1:** the robot encourages the user to move a token

**LEV\_2:** the robot suggests a subset of possible solutions

**LEV\_3:** the robot suggests the right token to move

**LEV\_4:** the robot grasps and offers the right token

The verbal assistance is provided on the base of Cutrona categorization [23]: information support (providing advice), tangible assistance (for example by providing the solution), esteem support (encouragement and motivation) and emotional support. To this end, at each level corresponds different sentences and movements, such as to have different ways to provide the same behaviour.

The assistive policy should increase the robot effectiveness and help to avoid disengagement due to lack or excess of assistance. The assistive behaviour from the robot is activated when the patient is expected to perform an action. We model the learning of the optimal assistive policy as a RL problem. Since the robot should learn through his experience while interacting with the patient, we endow the robot with a temporal difference RL algorithm that can be implemented to adapt online and it is model independent. Thus we propose to use a Q-learning method. In Q-learning, the policy is formulated as a  $Q(s; a)$  matrix, where  $s$  is the state of the environment in a given time and  $a$  the action the robot uses to shape the environment. In our scenario,  $s$  represents the current user's state, whereas  $a$  represents the one of the assistive action of the robot. The state-space consists of three dimensions: i) task progress, defined as the number of token sorted so far  $tp = \overline{1; \dots; Ng}$  where  $N$  is the task length; ii) attempts, defined as the number of attempts of the user on the current token  $att = \overline{1; \dots; Mg}$  where  $M$  is set the maximum number of attempts defined for the given task; iii) and robot assistance,  $lev = \overline{Lev.0; Lev.1; Lev.2; Lev.3; Lev.4g}$  defined as the level of assistance provided by the robot in the previous state. The robot learns its policy looking at the user's actions after it provides a supportive behaviour. At that stage the  $Q$  matrix is updated according to the standard equation:

$$Q(s; a) = (1 - \alpha) Q(s; a) + \alpha (r + \gamma \max_{a'} Q(s'; a')) \quad (1)$$

where  $s'$  is the new observed user's state after the action  $a$  is executed,  $\alpha$  is the learning rate and  $\gamma$  a discounting factor. The reward  $r$  value depends on the success of the given action of assistance provided by the robot.

$$r = \begin{cases} success & (tc \quad atp \quad lev) \\ fail & (tp \quad atp \quad lev) \\ max\_attempt & 100 \end{cases} \quad (2)$$

where,  $tc$  is the task complexity and is defined as  $(N - tp)$ ,  $lev$  is equal to  $(K - lev)$  (where  $K$  is equal to 5 that are the number of levels of assistance) and lastly  $atp$  is equal to  $(M - atp)$ .

The reward is formulated in a way to provide minimal but effective assistance in each given state.

## VII. EXPERIMENTS

To test PBS and to evaluate if our algorithm is able to learn different assistive policies starting from different user pro-

files, we propose two experiments. In the first experiment, we perform an evaluation on the effectiveness of the proposed Q-learning agent against a random agent interacting with four different Personas. In the second experiment, we analyse if the proposed algorithm is able to change its behaviour when the user's performances change over time. The four Personas are defined as follows (see Figure 2):

Paul: is a 85 years old man with severe dementia and Parkinson at late-stage.

Anne: is a 82 years old woman with moderate dementia and semi-paralysis on his left-part of the upper body.

Ben: is a 75 years old man with mild dementia

Rachel: is a 81 years old woman with no physical and mental impairment

The task is the one defined in Section VI-A. For simplicity,  $N$  is set to 5 and  $M$  to 4.

### A. Evaluate Robot Assistive Behaviour Generation

In this Section, we conduct an evaluation study to investigate to which extent the robot's behaviour can change and adapt to a patient profile generated from PBS. Therefore we evaluate it under two different conditions:

**random condition:** the robot selects the assistive actions according to a uniform distribution. In this setting, the robot does not take into account the user's actions and select its behaviour randomly.

**learning condition:** the robot decides the way to assist the user according to the state-action definition of Section VI-B. The state in which the user is at determines the amount of reward/penalty after an assistive action is provided by the robot. For this experiment, we use an  $\epsilon$ -greedy method for strategy selection, where  $\epsilon$  is the probability of taking random action and  $1 - \epsilon$  is the probability of exploiting the recommended action. According to [16], we set  $\epsilon$  equals to 0.4 so to guarantee a good action selection while for the learning rate  $\alpha$ , we choose 0.2 to balance the learning effect.

If we consider the cognitive training exercise as a task, we can evaluate the effectiveness of the assistance provided by the robot agent analyzing the number of attempts each patient profile needs to complete the exercise. We expect that the Persona in learning condition will solve the exercise faster than the same Persona when assisted by a random agent. Although the patients performances are an objective way to evaluate the capability of the designed agent, they might not be enough to establish if the provided assistance is the most suitable for a given user profile. Indeed, maintaining the highest level of assistance and independently from the subject capabilities, the outcome for most users will be to complete the exercises with a low percentage of mistakes. This behaviour must be avoided since our objective is both to improve user's performance and engagement. The balance between them has been proved to increase the user's commitment during the task and this is of vital importance to avoid user disengagement and guarantee long-term interaction. We address this issue defining the reward as in Equation 2. As

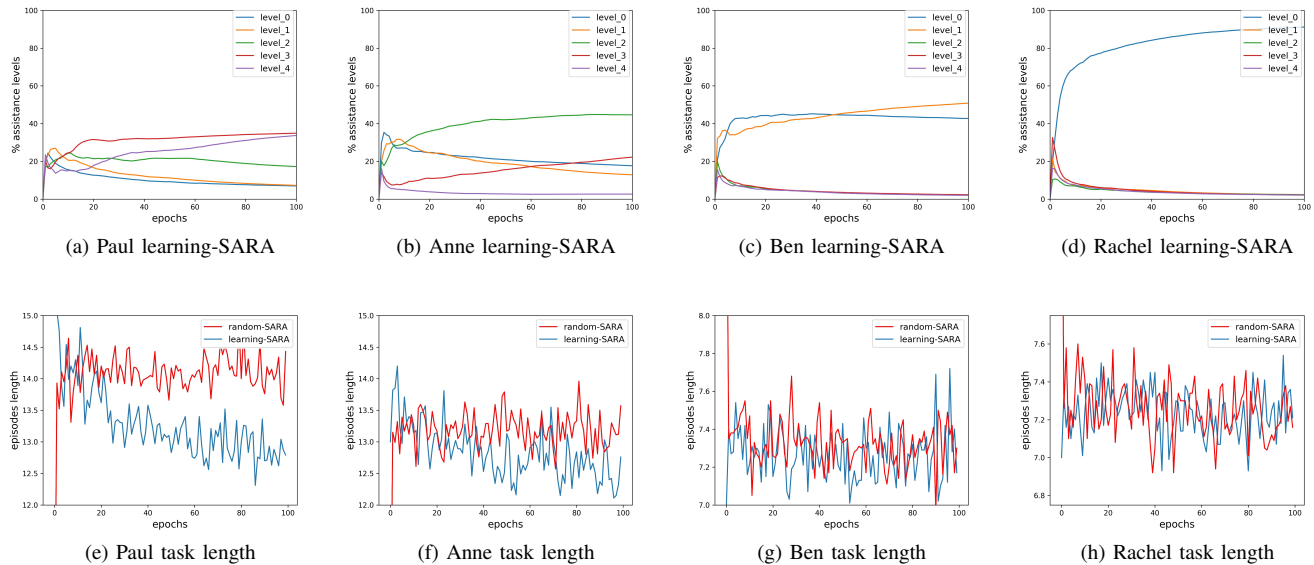


Fig. 4: In the first row we report the robot percentage of levels of assistance over epochs for each Persona profile (see Figure 2) while in the second row we report the performance of each Persona when a random-SARA (red line) and a learning-SARA (blue line) provide assistance.

it is can be observed, at the beginning of the exercise, the lower the assistance provided to the user, the harder the test and consequently higher will be the reward. On the contrary, at the last stages of the exercise, the higher the assistance provided to the user, the easier the test and consequently higher the penalty.

In Figures 4, we show the results of our simulation over 10,000 episodes for Paul, Anne, Ben and Rachel, respectively. We did not report the plots for the random-SARA since they are trivial. Random-SARA selects actions according to a random distribution where each action has a probability of 0.2 to be chosen. On the contrary, in the first row we analyze the behaviour of the learning-SARA. The Figures in the second row, instead, show the number of attempts performed by a user to solve the exercise when the assistance is provided by a random-SARA (red line) and by a learning-SARA (blue line). As it can be observed, the same Persona in learning conditions (1st row Figure 4) receives a completely different assistance by the robot compared to a random one. In the case of Paul (see Figure 4a), the agent detects after few iterations the struggle in completing the exercise and assist him 35% of the time with Lev\_3 and almost 35% with Lev\_4. Figure 4b show the assistance provided to Anne. Since her cognitive conditions are better than Paul, the agent provides her mainly with Lev\_2 (40% of the time) and Lev\_3 (20% of time) of assistance. According to our Persona profile, Ben is a patient with mild dementia that doesn't need so much assistance to solve the exercise (see Figure 4c). The agent behaviour reflects his characteristics. It offers him 45% of time Lev\_1, 40% of time Lev\_0 and the rest of the time assistance among Lev\_2 Lev\_3 and Lev\_4. Lastly, in the case of Rachel (see Figure 4d), the agent provides for most of the 85% of the time Lev\_0, which means only calling user's

attention to suggest a token to move.

An interesting analysis that deserves to be conducted is the comparison in term of user's attempts to solve the exercise under two heterogeneous conditions (Figures 4e-4h).

In Figure 4e we report the performance over time for Paul. Since his attention span is 2 (see Figure 2), most of the time the assistance provided by the robot agent is ignored. In addition, it is possible to appreciate how the learning-SARA (blue line), unlike the random-SARA (red line), is able to detect also a memory problem. In the case the focus is maintained, the robot agent keeps on giving the most suitable assistance to complete the test. The results shows clearly better performance in learning condition. Different is the case for Anne whose performances are not so evident when she is assisted by a learning-SARA as in the previous case. This is a behaviour that we will notice also with the others Personas Ben and Rachel. In these cases, the learning-SARA since their Personas skills are slightly better, in the case of Anne or much better in the case of Ben and Rachel, compare to Paul, it provides them with enough assistance to solve the exercise. This means providing a limited level of assistance. This is the reason why the performance of Ben and Rachel are almost the same when they are assisted from random-SARA and learning-SARA. In random conditions, the robot always provides for 60% of time high levels of assistance (20% Lev\_2, 20% Lev\_3 and 20% Lev\_4) no matter who is the Persona. So, although the two algorithms perform the same in the end, the random policy will not provide the user with the most suited assistance for his needs. This is a crucial requirement to take into account for an overall evaluation of the system. Moreover, the goal of the system is to provide an initial policy for the robot that will then be further personalized on the specific user thus providing the

robot with a reasonable initial policy will reduce the time for the robot to converge to an optimal policy.

### B. Evaluate learning-SARA in case of User’s Changing Behaviour

In this second experiment, we aim to evaluate the behaviour of the agent when the patient capabilities change over time and to which extent the agent is able to adapt to them. Since the user behaviour is not predictable and stationary we cannot assume that his performance are the same over time. Users responses are likely to change over time, and their behaviour is difficult to represent in a fix probabilistic manner. How we can keep the robot learning over time, taking into account that the user’s behaviour can suddenly change? In order to guarantee that our agent will not overfit over time on a given behaviour but on the contrary learning from the changes, we need to keep exploring during the epochs to be sure that the agent can always reshape its behaviour. We propose an adaptive formulation for  $\alpha$  and  $\beta$  values according to [20] that it gets more confident over time but differently from it, every  $n$  epochs we reset them in order to increase exploration and decrease the learning factor if the user’s behaviour changes. To prove the adaptability of the agent for eventual changes in user’s behaviour, we define a Persona with the following parameters: memory=3, attention=4, reactivity=4. We then suppose that the same Persona can: worse his performance over time, maintain his performance over time and finally improve his performance over time. We expect the agent to be able to learn three different policies in order to guarantee suited assistance for the user.

The results reporting the different levels of assistance provided by the agent in the three different cases are shown in Figure 5. Figure 5d shows the agent levels of assistance in case of the Persona worsening his performance over time. We can see how, even though on the basis of the initial conditions Lev\_2 appears to be the best supportive assistance, the agent starts providing the Persona with more support as soon as the performance deteriorates. That is the reason why Lev\_3 and Lev\_4 start increasing over time. On the contrary, a different agent’s behaviour can be noticed in Figure 5f. In that case, since the Persona starts performing better over time, Lev\_2 and Lev\_3 decrease while Lev\_0 and Lev\_1 increase.

In order to validate that the three different learned policies are effectively in a real scenario, we play with a fix strategy with three different agents. The results are reported in Table I. As it is possible to observe, when the agent plays with a Persona whose performance is decaying over time (Table I, 2nd column), the robot provides much more assistance in comparison with the static condition where the user is not supposed to change his behaviour (Table I, 3rd column). Moreover, the differences are even more evident if we compare them with the agent that assumes learning condition from the user (Table I, 4th column).

Lastly we compare the used strategy that resets periodically the adaptative  $\alpha$  and  $\beta$  (see Figure 5, 2nd row) against the static strategy (see Figure 5, 1st row) where  $\alpha$  and  $\beta$

user action	get worse	static	get better
X	Lev_3	Lev_3	Lev_1
1	Lev_4	Lev_4	Lev_4
X	Lev_3	Lev_3	Lev_2
2	Lev_4	Lev_3	Lev_0
X	Lev_2	Lev_2	Lev_2
3	Lev_3	Lev_2	Lev_2
X	Lev_3	Lev_3	Lev_2
4	Lev_2	Lev_1	Lev_1
5	Lev_2	Lev_1	Lev_0

TABLE I: Assistance provided by SARA when the user is deteriorating his performance over time (2nd col); maintaining the same performance over time (3rd col) and improving his performance over time (4th col). In gray it is highlighted the user’s  $N$  right move, while the cells with "X" are the attempts of the user before performing the right move.

set equal to 0.2 and 0.4 respectively at the beginning of the learning process. The objective is to evaluate to which extent the two different strategies affect the robot learning policy. As it is possible to notice, with a static strategy (see Figure 5, 1st row), the agent is not able to properly adapt to the user’s behaviour and after few iterations it already decides which are the preferred supportive assistance to provide the user. On the contrary, as we have already seen, with an adaptative strategy (Figure 5, 2nd row) the robot is able to adapt over time to the user’s changing behaviour.

## VIII. CONCLUSIONS AND FUTURE WORK

In this paper, we present a Persona-Behaviour Simulator (PBS) that generates high-level user’s actions for HRI tasks. The proposed simulator has been validated in a cognitive training scenario where, during simulated interactions, the Social Assistive Robot Agent (SARA) is able to learn from such actions its initial policy. We show that the learned policies are coherent with the user’s profile and that the proposed algorithm is also able to adapt when the user’s behaviour changes over time. Due to its high-level of abstraction of user’s actions, the PBS can potentially be used in most of the HRI scenarios where collecting data is difficult or not feasible. The proposed approach reduces the user burden, and his exposure to long/tiring training sessions, thus trying to minimize the number of trials the individuals have to perform.

Since the current PBS has been validated, the next step of our work is to make use of it defining patients profiles and the corresponding learned initial policies for real patients. This initial setting will be used in what we already defined as the second loop of interaction, in which through real interaction, the SARA will improve its initial policy to better meet the needs and preferences of the real user.

## ACKNOWLEDGMENT

We would like to thank Carla Abdelnour, Joan Hernandez and Natalia Tantinya from Fundaciò ACE for the fruitful discussions and the help in the design of the SKT-based exercise.

