2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Daejeon Convention Center
October 9-14, 2016, Daejeon, Korea

A Learning from Demonstration System Architecture for Robots Learning Social Group Recreational Activities

Wing-Yue Geoffrey Louie, *IEEE Student Member*, and Goldie Nejat, *IEEE Member*

Abstract— **Group-based recreational activities have shown to have a number of health benefits for people of all ages. The handful of social robots designed to facilitate such activities are currently only able to implement a priori known recreational activities that have been pre-programmed by human experts. Once deployed in their intended facility, these robots are not able to learn new activities from non-expert humans. In this paper, we present the development of a novel learning from demonstration (*LfD*) system architecture for a social robot in order for it to learn from non-expert teachers the structure of an activity and monitor the execution of the new activity. In order to obtain user compliance, personalized persuasive strategies are also learned by the robot to use while implementing the activity during human-robot interactions (HRI) with the intended users. The architecture has been integrated into our socially assistive robot Tangy to learn the group-based activity Bingo. System performance experiments were conducted with Tangy to first learn to facilitate Bingo from non-expert teachers and then use the learned activity to physically facilitate Bingo with multiple users. The results showed Tangy was able to effectively and efficiently learn the new Bingo activity structure as well as personalize its persuasive strategies to individual users in order to obtain activity compliance.**

I. Introduction

Group-based recreational activities (RAs) which involve participants socially engaging and supporting each other (e.g. clubs, team sports, games) have shown to have physical, social, emotional, and cognitive health benefits for everyone [1]-[3]. Therapeutic recreation aims to promote the positive benefits of social RAs to improve quality of life for individuals [4]. Namely, therapeutic recreation professionals support individuals in attaining their social recreational needs by providing such programs in the community including in healthcare facilities, schools, community centers and prisons.

Social robots are currently being developed to assist in providing group-based therapeutic recreation to users of all ages [5]-[7]. In [5], the child-like KASPAR robot was used to mediate turn-taking imitation games with children and adolescents with autism spectrum disorder to improve basic social interaction skills. Our own previous work in [6], consisted of using the social robot, Tangy, to facilitate Bingo with groups of adults. In [7], the seal-like robot Paro was used for group pet therapy for older adults with dementia to promote group social interaction.

The results of user studies with the aforementioned robots have shown the potential benefits of using robots for RAs. However, current social robots are limited to a set of a priori known activities that have been pre-programmed and integrated by human experts. Once deployed in their intended facility, these robots should be capable of learning new activities from non-expert humans to adapt to the needs of that facility. Namely, healthcare staff should be able to teach a robot new activities that are needed to effectively administer their therapeutic recreation programs, as well as personalize existing activities for intended users. This can improve user compliance and impact the health efficacy of such programs [8].

The objective of our research is to develop social robots capable of learning to autonomously facilitate group social RAs from non-experts. To achieve this goal, we have developed a novel learning from demonstration *(LfD)* robot architecture. The architecture is capable of: 1) learning activity structures from non-expert teachers and monitoring the execution of the new activity, and 2) personalizing robot assistive behaviors during the activity to promote compliance. Our goal is to design robotic technology that is adaptable and easy to use in the settings that robots are deployed in order to promote long-term use.

II. Activity Learning

In this section we discuss how *LfD* has been used to learn task/activity structures as well as how learning for personalization has been used during robot facilitated RAs.

A. Learning from Demonstration (LfD)

Robots can effectively learn new activity structures (i.e. high-level task representations) using *LfD*. In particular, two common *LfD* approaches that have been used include [9]: 1) a teacher teleoperating a robot to perform specific behaviors to accomplish a task [10]-[13], or 2) a robot using external observations of the teacher performing a task and then mapping these behaviors onto its own platform [14],[15].

For example, in [10], the PR2 robot learned from teleoperation-based kinesthetic demonstrations to flip over a box using two chopsticks. The teacher manipulated the robot's grippers to define the motion trajectory, which was used to learn the initial policy for the task. The learned policy was then optimized during task execution using reinforcement learning. During the demonstrations, sensor information was also used to predict task failures, which resulted in additional corrective demonstrations being provided to the robot for further learning. In [11], the Simon robot learned from teleoperation-based kinesthetic demonstrations to pour coffee beans from a cup into a bowl and to close the lid of the box. Speech commands were used to specify Keyframes that highlighted the important parts of a

*This research has been funded in part by the Natural Sciences and Engineering Council of Canada (NSERC), Dr. Robot Inc. and the Canada Research Chairs (CRC) program.

W.G. Louie and G. Nejat are with the Autonomous Systems and Biomechatronics Laboratory, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON, Canada (email: geoffrey.louie@mail.utoronto.ca, nejat@mie.utoronto.ca).

978-1-5090-3762-9/16/$31.00 ©2016 IEEE

task and kinesthetic demonstrations were used to demonstrate the motion to move between these Keyframes. Action and goal Hidden Markov Models were learned, from motion and object data, respectively, using a Baum-Welch algorithm. Tasks were executed using the learned action model, and failures were identified using the goal model to determine the probability of an observed state sequence.

In [14], an anthropomorphic robotic hand learned to grasp various objects from external observations of a teacher. A teacher demonstrated the initial hand pose to grasp an object while a neural network based method was used to identify teacher hand joint positions from stereo images and map the joint positions onto the robot hand. Potential grasping sequences were then generated by starting at the initial hand pose and closing the fingers until they were blocked by the object of interest. In [15], a mobile Pioneer 2-DX robot used external observations of a teacher to learn to perform various navigation and object delivery tasks. Namely, the robot tracked task related objects using a laser range finder and a 2D camera, to obtain a temporally ordered demonstration trajectory. Demonstration trajectories were then used in a longest common subsequence algorithm to learn high-level task structures. The robot also conducted practice trials in front of the teacher to obtain verbal feedback or add missed steps in the learned task sequence.

Current *LfD* approaches focus on providing robots with the ability to autonomously perform physical tasks. A handful of the aforementioned approaches ([11], [15]) also use social behaviors from the teachers to improve and facilitate demonstrations between the robot and a teacher (e.g. requesting for more demonstrations when the robot makes a mistake or it is not confident in performing the task). However, they have not focused on learning social tasks that involve direct social interactions between intended users and a robot [16]. Our problem differs in that we aim to have a robot learn a social activity which involves social interactions with a group of non-expert users.

B. *Learning for Personalization*

Studies have shown that social robots that personalize their interactions with users can improve the overall effectiveness of a facilitated RA [17], [18]. For example, in [17], the child-like robot, Bandit, changed the difficulty level of a music guessing game to improve user performance and maintain interest in the game. The robot used a supervised learning method based on a user's response speed and game success rate to personalize the robot's behaviors to change game difficulty. In [18], the human-like robot, Brian 2.1, adapted its assistive behaviors to reduce user stress levels during a memory card game. Brian 2.1 used a hierarchical reinforcement learning based method to learn appropriate assistive behaviors for a user to reduce stress levels.

The aforementioned HRI personalization techniques focused on adapting robot behaviors to improve user moods or performance. These learned personalized behaviors, however, were specific to the activity at hand, and were not generalized. Alternatively, in this work, we present a generic learning for personalization methodology that uniquely utilizes persuasive strategies [19] to gain user compliance during a facilitated activity. These persuasive strategies can be generalized to be used for different activities and personalized for each user to obtain compliance [20]. Obtaining user compliance is important during therapeutic RAs to promote cognitive and social health [8].

III. Demonstration Learning System Architecture

Our proposed demonstration learning system architecture is presented in Fig. 1. The objective of this system is to have a social robot learn the task representation of a social group RA and autonomously facilitate the activity with a group of users. The architecture is comprised of three sub-systems: 1) learning, 2) interaction, and 3) execution monitoring. Namely, the learning sub-system acquires activity demonstrations from non-expert teachers and learns the activity structure from these demonstrations. The interaction sub-system then uses the learned activity structure to have a social robot physically implement the activity with a group of users, as well as personalize these interactions. During the physical implementation of an activity, the execution monitoring sub-system detects and diagnoses faults as well as notifies the interaction sub-system to undertake recovery behaviors. These sub-systems are discussed in detail below.

A. *Demonstration Learning Sub-system*

Herein a teleoperation-based *LfD* approach is used by a non-expert teacher to demonstrate the group activity to the robot. Since we are focusing on activity-level learning, the robot has a set of known primitive behaviors and the goal for the teacher is to teach the robot a new activity that is not known a priori using these behaviors. For robot learning, an activity simulator is used by the teacher to represent the overall activity scenario including both the robot and the group of users. We have chosen to use a simulated environment to obtain demonstrations from non-experts as simulated environments improve the efficiency of learning and reduce teacher fatigue [21].

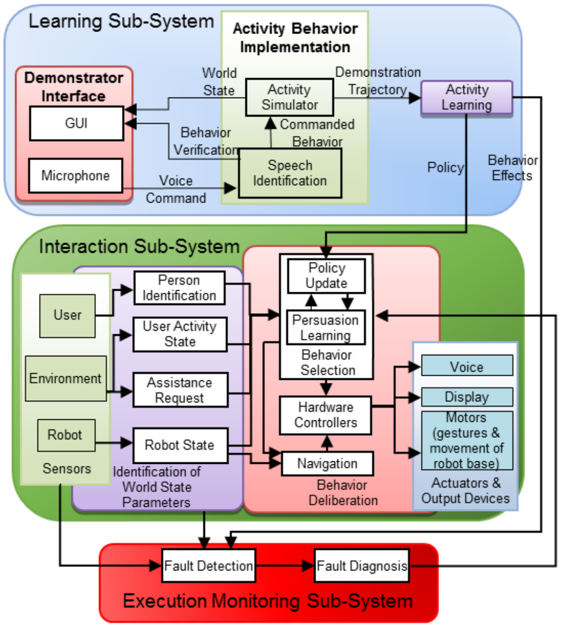


Figure 1. Demonstration Learning System Architecture

1) *Speech Identification*

A teacher uses speech commands spoken into a microphone to control the robot's behaviors in the activity simulator module to demonstrate the facilitation of an activity. Speech decoding takes place by utilizing the Sphinx speech recognition system [22]. A Hidden Markov acoustic model is used to label phonemes in the teacher's utterances and match labelled phonemes to words. An *n*-gram language model is then used to determine the sequence of spoken words. The identified sequence of words are then matched to a set of keywords associated with known robot behaviors.

2) GUI

The GUI is the primary interface for the teacher and presents the world state provided by the activity simulator. Namely, both the users and the robot are depicted as virtual agents and their behaviors are updated in real-time according to the teacher's commanded behaviors for the robot. Once the teacher provides a verbal command, he/she is then prompted through the GUI to verify if the identified robot behavior is to be executed by the robot. Namely, a pop-up message appears on the GUI stating, "Did you say [robot behavior]? Type 'Y' to execute or 'N' to provide a new command." The teacher provides this input through a keyboard to circumvent incorrectly identified commands from the speech identification sub-module.

3) Activity Simulator

The activity simulator module consists of models for the robot, group of users, and the activity.

i) Robot Model

The robot, *R*, is modeled as a simulated agent with a set of known behaviors, $B=\{b^1, b^2, \dots b^m\}$,where *m* is the total number of behaviors. Each behavior has a set of actions, $b^i = \{ac_1^i, ac_2^i, \dots ac_q^i\}$, where *q* is the total number of actions for behavior *i*. For example, the action ac_υ^i is action υ in the set of behavior *i* and is defined to be a function of robot actuator positions (*θ*), robot speech (*sp*), visual content displayed on the robot's screen (*im*), and desired robot location ($l_{r_{x,y}}$): $ac_v^i = f(\theta_v^i, sp_v^i, im_v^i, l_{r_{x,y},v})$.

ii) User Model

The users participating in the group activity are modeled as the set $U=\{u^1, u^2, \dots u^n\}$, where *n* is the total number of users participating in an activity at one time. Each individual user *r* is modeled as a set, $u^r = \{ID, s_{ua}, s_h, l_{u_{x,y}}\}$. Namely, *ID* is the name of the user; s_{ua} is the user's particular activity state; s_h is the user's assistance request state; and $l_{u_{x,y}}$ is the user's 2D location within the activity room.

iii) Activity Model

The activity is modeled as a set of activity states (e.g. start, facilitate, help, socialize, and end) during the facilitated group activity, $S_a = \{s_a^1, s_a^2, \dots s_a^g\}$, where *g* is the total number of activity states that occur during the entire activity. Each activity state is defined by $s_a^p = \{k, s_{ua}^{1,2\dots n}, s_h^{1,2\dots n}\}$, where *p* is an instance of the activity state and *k* is the discrete time step.

iv) Trajectory

At the end of an activity demonstration, the sequence of behavior-state pairs is used to define the trajectory for the robot. Namely, the trajectory can be defined as $T = \{b^1, s_a^1\} \rightarrow \{b^2, s_a^2\} \dots \rightarrow \{b^j, s_a^j\}$, where *j* is the total number of state-behavior steps required for the complete demonstration.

4) Activity Learning

Within the activity learning module the state-behavior mapping (i.e., policy) of an activity is learned. We utilize a random forest classifier to learn this policy. Random forest is chosen as it can provide a confidence on its learned policy and avoids overfitting to state-behavior pairs in the trajectory [23]. In our learning scenario, we utilize the trajectory *T* as the training set to learn the policy for an activity. Namely, tree-structured classifiers are generated from independent identically distributed state-behavior pairs, $\{b^i, s_a^p\}$, sampled from the trajectory. Robot behaviors b^i are considered the classes and the activity states s_a^p are the features in the training samples. The set of generated trees defines the policy function, $\pi(s_a^p) = b^i$, which provides a state-behavior mapping for the demonstrated activity. Namely, input activity states are classified by each generated tree casting a vote for the appropriate behavior to execute. The behavior that receives the most votes is then mapped to the input activity state. The policy function is then utilized by the behavior selection module in the interaction sub-system to facilitate the demonstrated activity.

A robot needs to know how its executed behavior affects the activity to allow the robot to monitor whether this behavior is achieving the intended goal. In our learning scenario, the robot does not know a priori the effects of its behaviors within the context of a demonstrated activity. Instead the robot assumes the intended effects of a behavior is the activity state that immediately follows the execution of a behavior during an activity demonstration. Namely, our behavior effects identification algorithm parses the demonstration trajectory into sets, $\{b^i, s_a^{k+1}\}$, where b^i is the executed behavior and s_a^{k+1} is the observed activity state one time step after executing the behavior. These sets are then used to define the expected effects of each robot behavior in a demonstrated activity. The expected robot behavior effects are then used in the fault detection module in the execution monitoring sub-system to detect for occurrences of faults during task execution.

B. Interaction Sub-system

The interaction sub-system uses the learned activity policy to have the robot autonomously facilitate group RAs in the real-world with the intended users. This is achieved by implementing the robot's physical behaviors based on sensed world states. During activity facilitation, the robot also adapts and personalizes its behaviors to obtain user compliance using persuasive strategies that influence changes in attitudes and behaviors.

1) Identification of World State Parameters

Sensory information of the environment, users and robot are obtained to determine world state parameters. These parameters include each user's, *r,* identity *ID* determined from user sensors; and activity state s_{ua}, user assistance request state s_h and 2D location $l_{u_{x,y}}$ within the activity room all determined from environmental sensors.

Furthermore, the robot's 2D location $l_{r_{x,y}}$ is determined by the robot sensors. These observed parameters are used to define the activity state s_a^p in the behavior selection module.

2) Behavior Deliberation

The behavior deliberation modules determine the robot's behavior b^i and associated action ac_v^i to execute based on the observed activity state s_a^p. The associated action ac_v^i for each behavior b^i has been designed to also include a persuasive strategy. Namely, the robot has a set of persuasive strategies, $PS = \{ps_1,...,ps_\gamma\}$, where γ is the total number of robot persuasive strategies. Namely, each action is updated to also include the persuasive strategy, ps_z, utilized during HRI: $ac_v^i = f(\theta_v^i, sp_v^i, im_v^i, l_{r_{x,y},v}, ps_{z,v}^i)$.

Behavior Selection: The behavior selection module first determines the appropriate robot behavior b^i to implement by utilizing the learned activity policy π. The appropriate associated persuasive strategy is then determined based on user learned persuasive strategy profiles during activity facilitation. We define a user's persuasive strategy profile as the set, $M = \{\Omega_{ps_1},...,\Omega_{ps_\gamma}\}$, where Ω_{ps_z} is the probability of complying to a particular persuasive strategy ps_z. User persuasive strategy profiles M are learned using a Thompson Sampling based approach [24]. Thompson Sampling is used to learn the model for the user persuasive strategy profiles by maintaining a belief on the expected probability of a user r complying with each ps_z based on previous interactions with that user. Persuasive strategies are then selected by sampling from these beliefs for the ps_z that provides the highest probability of user r complying during an interaction. Beliefs are then updated according to the success of the executed ps_z. Namely, Thompson Sampling determines when to explore or exploit different persuasive strategies. In our Thompson Sampling based learning approach, we assume for user r each Ω_{ps_z} has a Beta distribution:

$$\Omega_{ps_z} = f(x;\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \quad 0 \le x \le 1 \quad , \qquad (1)$$

where x is the probability that user r will comply with ps_z, and α and β are initially set to one. The decision to exploit or explore strategies is then determined by sampling from the modeled Beta distributions of Ω_{ps_z} for all persuasive strategies in M. Namely, the sampled strategy that provides the highest probability of compliance is selected. After execution of a selected strategy, the Beta distribution for the selected strategy is updated according to the following rule:

$$\alpha,\beta = \begin{cases} \alpha+1, \;\; \beta, & \text{if user complied} \\ \alpha \;\; , \;\; \beta+1, & \text{if user did not comply} \end{cases} \quad . \qquad (2)$$

Once ps_z is selected, the action ac_v^i associated with behavior i is sent to the hardware controllers of the robot to execute the appropriate actuator and output device commands.

Navigation: If ac_v^i involves the robot moving from one location to another location, the navigation module is notified to determine an appropriate path for the robot. The path is then provided to the hardware controllers to execute the appropriate actuator commands.

C. Execution Monitoring Sub-system

The objective of the execution monitoring sub-system is to ensure the successful implementation of the activity by the robot. Namely, to identify if a robot's behavior cannot be implemented due to the occurrence of a fault. Herein, a fault is defined as an event that causes the robot to transition to an incorrect or unknown activity state when implementing a behavior. The fault detection module uses information about the current activity state and the learned behavior effects to determine if a fault has occurred, while the diagnosis module uses information from all the sensors and world state identification module to determine fault recovery behaviors.

The fault detection module identifies faults by utilizing the identified behavior effects, $\{b^i, s_a^{k+1}\}$, from the Activity Learning module. Namely, after each behavior is executed by the robot, the activity state, s_a^p, is compared to the identified effects of the robot behavior, s_a^{k+1}. When the states do not match, a fault is identified and the robot queries the fault diagnosis module to determine the cause of the fault.

The fault diagnosis module uses a rule-based reasoning approach to analyze the cause of the fault. The criticality of the fault (non-critical versus critical) is also determined as a function of the type of fault. An example critical fault would be if the robot base cannot move due to hardware malfunction. Therefore, an expert is required to fix the problem. Non-critical faults can be furthered categorized as: 1) faults that will still allow the robot to continue the activity, and 2) faults that result in the robot requesting for assistance from a user (e.g. move activity object) or teacher (e.g. provide a new demonstration).

IV. LEARNING TO FACILITATE BINGO GAMES

We have integrated our *LfD* system architecture with our socially assistive robot, Tangy, to learn to facilitate the group-based game of Bingo. Namely, Tangy learns from non-expert teachers the activity structure of Bingo games and facilitates the learned game with a group of users. We chose Bingo as it has been shown to be effective in improving cognitive skills such as recognition, recall, and visual search as well as promotes social engagement among players [25].

A. Tangy Robot

Tangy, Fig. 2(a), has a human-like upper torso and a differentially driven wheeled mobile base. Namely, Tangy's upper torso consists of two six degrees-of-freedom (DOF) arms with two DOF grippers. Mounted on top of Tangy's torso is a six DOF animated head with one DOF for opening and closing the mouth, one DOF for each eye that allows for panning left and right, one DOF for moving the eyes up and down together, and two DOF in the neck for nodding and shaking the head. A tablet is mounted on Tangy's chest for displaying written messages or images related to the activity. The robot can also communicate verbally using a synthesized voice. The robot retrieves activity, user, and environment information using a combination of sensors including a laser range finder mounted on its base, 2D cameras mounted on its head and for its eyes, and an IR sensor placed in the activity room behind Tangy.

B. Bingo Scenario

The robot facilitated Bingo game scenario begins with Tangy at the front of a room facing four users seated behind a table. Each user is given a Bingo card that consists of a 5x5 grid of numbers from 1-75. During the game, Tangy will call out Bingo numbers and the goal is for the users to mark these numbers on their cards with red circular markers. At any point during the game, the users can request for assistance from the robot by pressing the button on their assistance request device (Fig. 2(b)), at which time Tangy will approach them to provide one-on-one assistance. A player wins the game when he/she has marked a row, column, or diagonal on his/her card correctly.

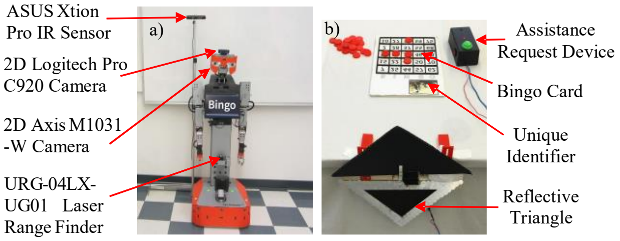


Figure 2. a) Tangy; b) Bingo card and user' assistance request device

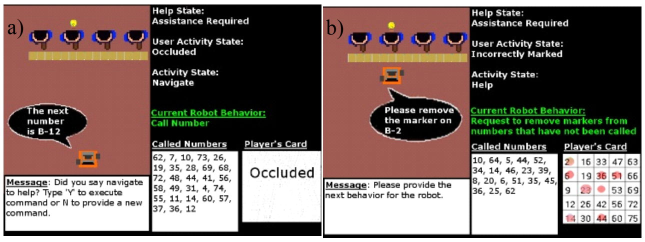
Tangy does not know a priori the Bingo activity structure. The robot has a known set of prior primitive behaviors (we obtained from our previous work [6]). The list of these behaviors can be found in Table II under the actual robot behavior column. For our scenario, Tangy uses four persuasive strategies to define the robot's actions: praise, suggestion, scarcity, and neutral. We use these strategies as they have been shown to be effective in HCI requiring user compliance [19]. Namely, offering praise and suggestions at opportune conditions have been shown to influence user compliant behaviors, and users find opportunities more valuable when they are less available [19]. We also include a neutral strategy with no social influence, as some individuals can be resistant to the above strategies [26]. Tangy changes its speech content for each strategy. Examples of Tangy's speech for persuasive strategies are presented in Table I.

Table I. Example Persuasive Strategies

Persuasive Strategy	**Example (Robot Behavior)**
Neutral	The next number is B-5. **(Call Bingo number)**
Praise	You are doing great, but have missed marking the following numbers: B-2, B-4 **(Request to mark numbers that have been called)**
Suggestion	You are about to win the game but have marked some incorrect numbers. Please remove markers from the following numbers. B-3, B-5 **(Request to remove markers from numbers that have not been called)**
Scarcity	Everyone is close to having a winning card. To increase your chances of winning, push your Bingo card closer to me so I can check if you have won. **(Request to move card closer to robot)**

C. Bingo Simulator

The Activity Simulator used for the Bingo scenario is presented in Fig. 3. As can be seen in the figure, Tangy and the users are simulated herein using the aforementioned Bingo scenario in order for a teacher to demonstrate the Bingo activity to the robot.



a)

The next number is B-12

Help State:
Assistance Required

User Activity State:
Occluded

Activity State:
Navigate

Current Robot Behavior:
Call Number

Called Numbers
62, 7, 10, 73, 26, 19, 35, 28, 69, 68, 72, 48, 44, 41, 56, 58, 49, 31, 4, 74, 55, 11, 14, 60, 57, 37, 36, 12

Player's Card
Occluded

Message: Did you say navigate to help? Type 'Y' to execute command or N to provide a new command.

b)

Please remove the marker on B-2

Help State:
Assistance Required

User Activity State:
Incorrectly Marked

Activity State:
Help

Current Robot Behavior:
Request to remove markers from numbers that have not been called

Called Numbers
10, 64, 5, 44, 52, 34, 14, 46, 23, 39, 8, 20, 6, 51, 35, 45, 36, 25, 62

Player's Card

2	16	33	47	63
6	19	36	51	66
9	23	●	53	69
12	26	42	56	72
14	30	44	60	75

Message: Please provide the next behavior for the robot.

Figure 3. Activity Simulation during a demonstration of a Bingo game: a) robot calls a number and user requests for assistance; b) robot requesting a user to remove marker from uncalled number.

D. Bingo Games with Users

Tangy implements the Bingo game scenarios in the real-world with human users using the learned activity policy. Namely, Tangy uses its sensors and world state identification modules to: 1) identify user activity states with the 2D camera mounted on its head, 2) determine the identity of users with the 2D camera in one of its eyes, 3) monitor when a user has requested for assistance with the IR sensor mounted in the activity room, and 4) localize itself within the RA room with the laser range finder mounted on its base.

User activity states are identified by determining the numbers marked on a user's Bingo card. Namely, a Hough transformation [27] based methodology is used to identify grid squares. A speeded-up robust features (SURF) [28] based methodology is then used to identify the unique identifier on the card (which has a corresponding set of Bingo numbers). Red blob detection is then used to identify which of these numbers are marked. User identities are determined by recognizing eyebrow, eye, nose, mouth, and face contour features via the OKAO^{TM} Vision software library [29]. User assistance requests are monitored using the IR sensor and a Hough Transformation [27] based methodology to identify IR reflective triangles that are revealed when a user presses his/her assistance request device, Fig. 2(b). The location of the user requesting for assistance is then determined by identifying the position of the IR triangle in the 3D point cloud of the environment. The robot localizes and navigates itself in the room using the laser range finder mounted on its base and optical encoders used with its base motors. Namely, Tangy uses the ROS navigation software package [30] to perform autonomous navigation with the aforementioned sensors. Details about Tangy's sensors and detection methods can be found in [6].

V. EXPERIMENTS

Two types of performance tests were conducted to verify the performance of our proposed *LfD* system architecture.

Scenario 1 - The performance of the learning sub-system in learning the structure of a Bingo game from non-expert demonstrations was investigated by determining the following: 1) the minimum number of Bingo demonstrations required to learn the activity policy, and 2) if different teachers have any effect on the learned policy. The non-expert teachers who participated in the experiments were university students with no previous robot teaching experience. First, one non-expert teacher demonstrated the group activity of Bingo to Tangy. We asked the teacher to incrementally provide demonstrations until we identified the minimum number of demonstrations required by the activity learning module to learn the expected policy. Five different non-expert teachers then demonstrated the Bingo activity to Tangy using this minimum number of demonstrations. The learned policies from each teacher were then compared.

Scenario 2 – The performance of the persuasion learning approach was investigated by determining the percentage of optimal persuasive strategy instances occurring during interactions with users. The optimal persuasive strategy was defined to be the strategy with the highest probability of a user complying with a request. Herein, we define interactions as situations during Bingo where users need to comply with the robot's requests. In this scenario, we used the activity simulator to investigate the ability of the persuasive learning approach to converge to an optimal persuasive strategy for four users having varying persuasive preferences. Bingo games were facilitated using a policy learned from one of the teachers in Scenario 1 within the interaction sub-system. The robot then learned user persuasive strategy profiles during the activities and adapted its persuasive strategies to maximize user compliance.

Scenario 3 – Experiments with Tangy physically facilitating twenty Bingo games with four real users were then conducted to verify the effectiveness of the proposed methodology. The same policy as in Scenario 2 was used. Users were university students (different from the teachers). During the activity facilitation, we also tested the execution monitoring sub-system by inducing faults (e.g. occluding and disconnecting sensors, and blocking the robot's path when navigating).

A. Results & Discussions

Scenario 1 Results: It took on average 9.6 minutes to teach a complete Bingo game to Tangy. This is three times faster than the time it took to play an individual game that was physically implemented by Tangy with a group of players (i.e., a physical game takes approximately 30 minutes). An average of 62 executed robot behaviors were implemented by each teacher during the learning stage. It was determined that three demonstration games were needed as the minimum number of demonstrations required to learn the policy. Three game demonstrations were required as not every help scenario was represented in every game in the simulator, as the game scenarios were randomly generated. We verified that if all scenarios were present in a single game demonstration, then the policy was able to be effectively learned in one demonstration. As expected, the exact same activity policy was obtained by the five teachers.

Scenario 2 Results: On average each user had 1.13 interactions per game. Fig. 4 presents the percentage of optimal strategy instances that occurred with each user across all interactions. On average 10.75 sub-optimal persuasive strategies were explored by the robot, before convergence to the optimal persuasive strategy for each user was achieved. The robot then only exploited the persuasive strategy with the highest probability of a user complying.

Scenario 3 Results: The results of the real-world Bingo game interactions are presented in Table II. The users always followed the behaviors of Tangy (but not necessarily the persuasive strategies). Tangy was able to successfully select and execute its activity behaviors for the 20 Bingo games using the learned policy. The robot was also able to learn user persuasive strategy profiles during activity facilitation. Furthermore, the robot was able to determine the appropriate recovery behaviors based on the induced faults. For example, when the robot identified that it couldn't help a user because an obstacle was placed in front of the table; it requested assistance from the user to remove the obstruction. The robot also requested assistance from an expert (one of the researchers) when it diagnosed that the 2D camera on its head was disconnected and it could not sense the Bingo card.

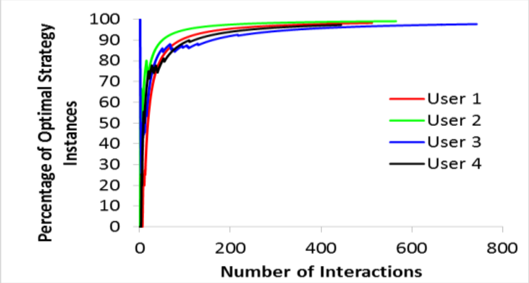


Figure 4. Percentage of optimal persuasive strategy instances for each user

VI. Conclusion

In this paper, we propose a novel *LfD* system architecture for a social robot to learn new group activities from non-experts and personalize interactions with users to obtain compliance. The system architecture allows non-experts to demonstrate group activities through a simulator that models the activity. From these demonstrations, the architecture learns a policy to facilitate a new group activity. During the facilitation of an activity, the robot also learns user persuasive strategy profiles and determines user persuasive strategy preferences to obtain compliance. The proposed architecture was implemented and tested for our social robot Tangy to autonomously facilitate Bingo games with multiple users. System performance experiments showed that the activity policy learned can successfully be implemented on the robot and the persuasive learning approach can be used to personalize interactions to obtain compliance from a user. For our future work, we will expand this work to include the use of *LfD* methods to also have the robot learn primitive behaviors. We will also be conducting pilot studies at long-term care facilities.

Acknowledgment

The authors would like to thank Sharaf Mohamed and Andrew Bowers for their assistance with this work.

References

[1] A. Passmore, "The occupation of leisure: Three typologies and their influence on mental health in adolescence," OTJR: Occupation, Participation and Health, vol. 23, no. 2, pp. 76-83, 2003.

[2] M. Crowe et al., "Does participation in leisure activities lead to reduced risk of Alzheimer's disease? A prospective study of Swedish twins," J. Gerontol. B Psychol. Sci. Soc. Sci., vol. 58, no. 5, pp. 249-255, 2003.

[3] G. N. Siperstein et al. "Social inclusion of children with intellectual disabilities in a recreational setting," Intellect. Dev. Disabil., vol. 47, no. 2, pp. 97-107, 2009.

[4] M. Carter et al., "The Nature of Therapeutic Recreation," Therapeutic recreation: A practical approach, USA: Waveland Press, 2011.

[5] B. Robins et al., "From isolation to communication: a case study evaluation of robot assisted play for children with autism with a

minimally expressive humanoid robot," IEEE Int. Conf. Adv. Comput-Hum. Interact., pp. 205-211, 2009.

[6] W. Louie et al., "An autonomous assistive robot for planning, scheduling and facilitating multi-user activities," IEEE Int. Conf. Robotic. Automat., pp. 5292-5298, 2014.

[7] W. Moyle et al., "Exploring the effect of companion robots on emotional expression in older adults with dementia: a pilot randomized controlled trial," J. Gerontol. Nurs., vol. 39, no. 5, pp 46-53, 2013.

[8] I. M. Rosenstock, "Patients' compliance with health regimens," J. Am. Med. Assoc., vol. 234, no. 4, pp. 402-403, 1975.

[9] B. D. Argall et al., "A survey of robot learning from demonstration," Robot. Auton. Syst., vol. 57, no. 5, pp. 469-483, 2009.

[10] P. Pastor et al., "Skill learning and task outcome prediction for manipulation," IEEE Int. Conf. Robot. Automat., pp. 3828-3834, 2011.

[11] B. Akgun et al., "Simultaneously learning actions and goals from demonstration," Auton. Robot., pp. 1-17, 2015.

[12] S. Niekum et al., "Learning grounded finite-state representations from unstructured demonstrations," Int. J. Robot. Res., vol. 34, no. 2, pp. 131-157, 2015.

[13] S. Chernova et al., "Interactive policy learning through confidence-based autonomy," J. Artif. Intell. Res., vol. 34, no. 1, pp. 1-25, 2009.

[14] J. J. Steil et al., "Situated robot learning for multi-modal instruction and imitation of grasping." Robot. Auton. Syst., vol. 47, no. 2, pp. 129-241, 2004.

[15] M. Nicolescu et al., "Learning and interacting in human-robot domains," IEEE T. Syst. Man. Cyb., vol. 31, no. 5, pp. 419-430, 2001.

[16] W. B. Knox et al., "Learning social interaction from the wizard: A proposal," AAAI Conf. Artif. Intell., pp. 1-6, June 2014.

[17] A. Tapus et al., "The use of socially assistive robots in the design of intelligent cognitive therapies for people with dementia," IEEE Conf. Rehabil. Robot., pp. 924-929, 2009.

[18] J. Chan et al., "Social Intelligence for a Robot Engaging People in Cognitive Training Activities," J. Adv. Robot. Syst., vol. 9, no. 113, pp. 1-13, 2012.

[19] B. J. Fogg, Persuasive technology: using computers to change what we think and do, San Francisco, USA: Morgan Kaufmann, 2003.

[20] M. Kaptein et al., "Selecting effective means to any end: Futures and ethics of persuasion profiling," in Persuasive technology, Berlin, Heidelberg: Springer, 2010, pp. 82-93.

[21] J. Aleotti et al., "Leveraging on a virtual environment for robot programming by demonstration," Robot. Auton. Syst., vol. 47, no. 2, pp. 153-161, 2004.

[22] Carnegie Mellon University, "CMU Sphinx", Available: http://cmusphinx.sourceforge.net/ [Accessed 2016].

[23] L. Breiman, "Random forests," Machine learning, vol. 45, pp. 5-32.

[24] O. Chapelle et al., "An empirical evaluation of thompson sampling," Adv. Neural. Inf. Process. Syst., pp. 2249-2257, 2011.

[25] B. P. Sobel, "Bingo vs. physical intervention in stimulating short-term cognition in Alzheimer's disease patients," Am. J. Alzheimers Dis. Other Demen., vol.16. no.2, pp.115-120, 2001.

[26] M. Friestad et al., "The persuasion knowledge model: How people cope with persuasion attempts," J. Consum. Res., vol. 1, pp. 1-31, 1994.

[27] R. Duda et al., "Use of the Hough Transformation to Detect Lines and Curves in Pictures", Commun. ACM, vol. 15, pp. 11-15, 1972.

[28] B. Herbert et al., "Speeded-Up Robust Features (SURF)", Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346-359,2008.

[29] Omron, "OKAO Vision", Available: www.omron.com [Accessed 2016], 2007.

[30] E. Marder-Eppstein et al., "The office marathon: Robust navigation in an indoor office environment", IEEE Int. Conf. Robot. and Automat., pp. 300-307, 2010.

TABLE II. EXECUTED ROBOT BEHAVIORS DURING IMPLEMENTATION

Actual Activity State	Actual Assistance Request State	Actual User Activity State	Actual Robot Behavior	Success Rate	Total Instances of Activity State
Start	ANR	Occluded	Greeting	100%	20
Socialize	ANR	Occluded	Joke	100%	29
Facilitate	ANR	Occluded	Call Bingo number	100%	699
Help	AR	Bingo	Celebrate	100%	20
Help	AR	Incorrectly Marked	Request to remove markers from numbers that have not been called	100%	26
Help	AR	Missing Numbers	Request to mark numbers that have been called	100%	23
Help	AR	Correctly Marked	Encourage user to keep up the good work	100%	18
Help	AR	Occluded	Request to move card closer to robot	100%	23
Navigate	AR	Occluded	Navigate to user	100%	71
Navigate	ANR	Occluded	Navigate to front of room	100%	36
End	AR	Occluded	Valediction	100%	20
Actual Fault Criticality			**Actual Robot Recovery Behavior**	**Success Rate**	**Total Instances**
Critical			Request Assistance from Expert	100%	21
Non-Critical			Continue Activity	100%	40
			Request Assistance from User	100%	6
			Request Assistance from Teacher	100%	20

*ANR = Assistance not required, AR = Assistance Required

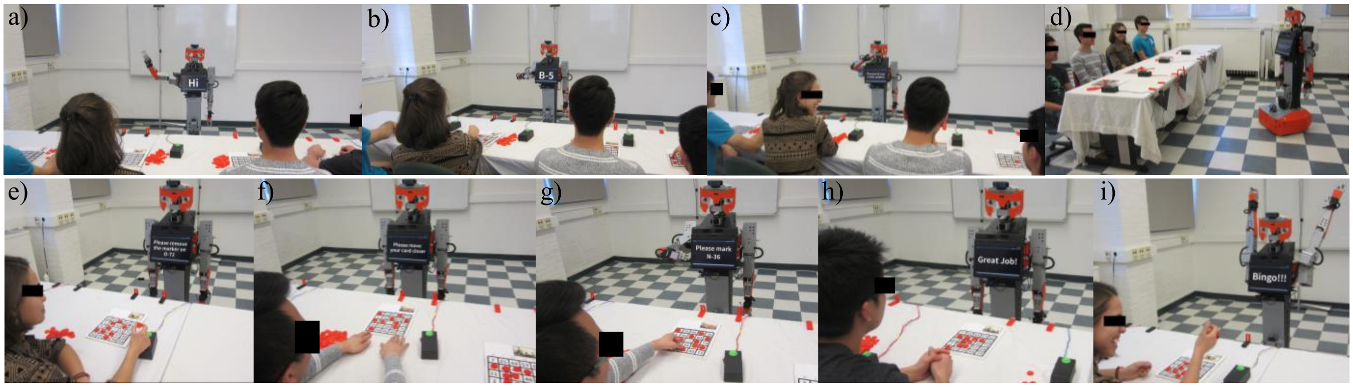


Figure 5. Robot Behaviors: a) Greeting; b) Call Number; c) Joke; d) Navigate; e) Request to remove markers from numbers that have not been called; f) Request to move card closer to robot; g) Request to mark numbers that have been called; h) Encourage user to keep up the good work; and i) Celebrate.