# TOWARD AGENTS THAT CAN LEARN NONVERBAL INTERACTIVE BEHAVIOR

*Yasser Mohammad and Toyoaki Nishida*

Kyoto University
Graduate School of Informatics
Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501 Japan

## ABSTRACT

Humans are social agents and the social dimension is an important aspect of human cognition. One challenge facing the realization of artifacts and artificial agents that posses human-like cognition abilities is to implement human-like interactive capabilities into them. Natural Language Processing is one of the earliest applications of AI techniques because of the importance of language in shaping human cognitive and interactive capabilities. Nevertheless nonverbal communication is starting to gain more importance specially in the domains of HRI and ECA because natural human-human communications are known to utilize a variety of nonverbal interaction protocols. This paper proposes a new adaptation algorithm for interactive agents that aims to develop agents that can learn and adapt their theory of mind concerning nonverbal interaction in real-time during actual interactions. The proposed method utilizes elements of the theory of theory and the theory of simulation to guide the adaptation process. A proof of concept simulation experiment with the proposed system is also illustrated.

***Index Terms*—** Social Cognition, EICA, Learning Interactive Behavior, Interactive Adaptation.

## 1. INTRODUCTION

The social dimension of cognition is considered by some researchers as one of the most important dimensions in shaping human cognition. In the recent years many researchers in HRI started to focus on implementing natural interaction modalities including nonverbal interactions into robots [1],[2]. Nevertheless most of the approaches involves hand coding of the interactive behavior. One reason for this situation is that intuitive and natural interactive behavior is not well specified like other kinds of behavior the agent needs to achieve and furthermore, personality and social backgrounds affect nonverbal interactive behaviors in humans [3] which makes it very difficult to come up with accurate models of the partner.

This paper is a first effort toward realizing an agent that can learn the interaction protocol by engaging in interactions with other intelligent agents (including humans). The following section presents the main architecture of the proposed system followed by detailed explanation of the algorithm used to learn the nonverbal interaction protocol in section 3. Section 4 presents a proof of concept experiment that provides some insight about the applicability of the proposed system. In section 5 a brief discussion of the relation between the proposed system and some influential architectures for interactive agents. The paper is then concluded with a discussion of the limitations of the current approach and future directions of research.

This work is partially funded by the Global Center of Excellence Young Leaders Fostering Program

## 2. THE PROPOSED ARCHITECTURE

Fig. 1 shows a simplified version of the proposed system which is implemented on top of the Embodied Interactive Control Architecture (EICA) proposed by the authors in [4]. The goal of the proposed system ($L_i$EICA) is to build agents that can achieve three main objectives:

1. Learning the structure of natural human-human interactions autonomously by watching those interactions in the real world.
2. Applying the learned structure to human-agent interactions to achieve a higher level of naturalness during those interactions.
3. Adapting the initially learned structure/protocol online during its own human-agent interactions.

This paper will focus on how the proposed architecture can achieve the last two objectives assuming that the first objective is already met. The proposed solution is motivated by multiple disciplines including nonverbal interaction studies, developmental psychology and neuroscience.

To achieve natural interaction with humans, the agent needs to synchronize its behavior with the behavior of the human at multiple time scales using different kinds of process ranging from deliberative role switching to reactive body alignment. For example research in nonverbal interaction studies found a rich set of synchronization protocol that utilize proximities, body alignment, nonverbal sound synchrony and other entrainment phenomena at different levels. Researchers in this

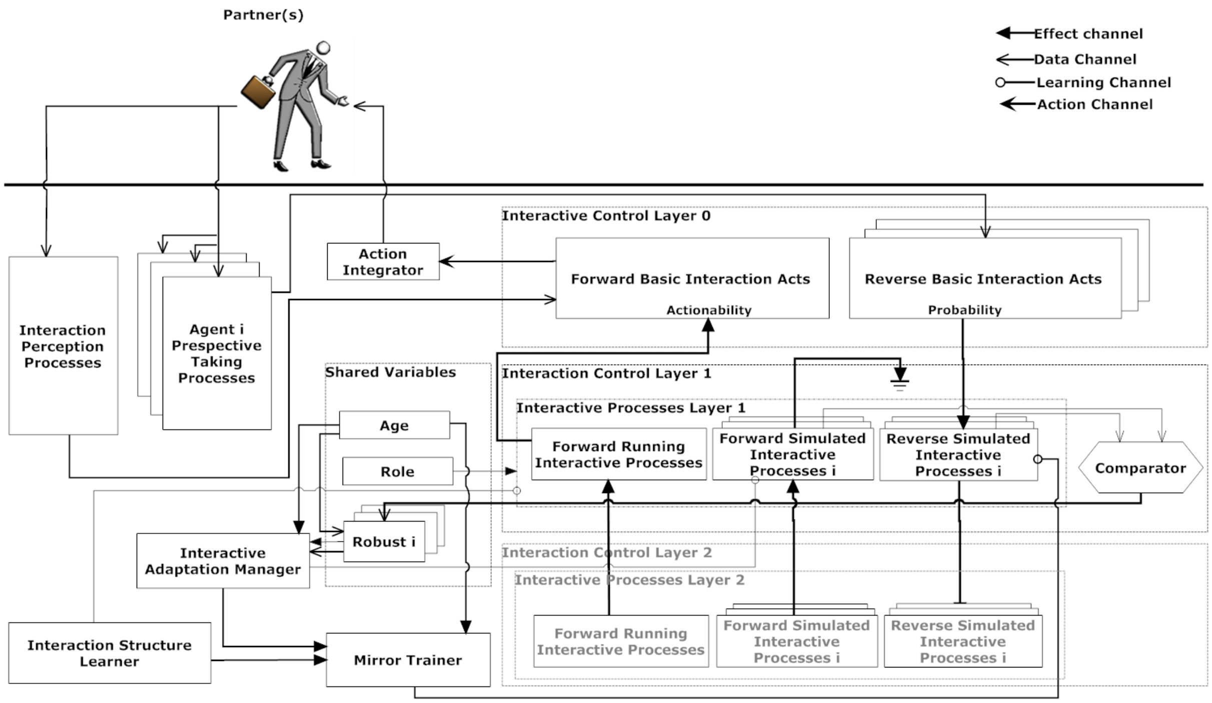


**Fig. 1**. A simplified version of the proposed architecture

area suggested that this synchrony happens in time scales that go down to hundreds of milliseconds in what was called the gestural dance [3].

The proposed system supports this natural layered feature of human-human interaction by structuring the control processes of the agent in multiple layers called interaction control layers. Within each layer a set of interactive processes provide the protocol needed to synchronize the behavior of the agent with the behavior of its partner(s) at a specific timescale based on a global *role* variable that specifies the role of the agent in the interaction.

The structure of the interactive processes in each layer is learned incrementally using the Interaction Structure Learner (ISL) in the first stages of the agent's development by watching other agents playing different roles. The work presented in this paper assumes that this stage have already passes leaving the agent with a set of $n^l$ processes in each layer each of which is known up to a parameter vector $p_i^l$ where $l$ represents the layer number and $i = 1 : n^l$.

The perceptual subsystem consists of a set of sensors that read relevant signals from the environment and other partners. Those signals are then processed by two sets of processes:

1. Perception Processes: a set of $n_s$ processes each of which detects a feature in the sensed signals. All features are represented as real numbers and combined in the situation vector (${}_c\hat{s}$).

2. Perspective Taking Processes. A set of $n_s \times n_a$ processes each of which detects one of the features from the perspective of one of the interacting partners. $n_a$ is the number of partners involved in the interaction.

The behavior of the agent is governed by a set of processes arranged in layers as shown in Fig. 1. Every process in the system has a set of inputs a set of outputs and an activation level. Processes with nonpositive activation levels are not allowed to run while the actuation commands issued by active processes are combined by weighting them with the activation level of the source process. The processes at the first layer are called basic interaction acts (BIAs) and the processes in higher layers are called interaction control processes (ICPs). Every BIA or ICP consists of two twin processes: the forward process -${}_i^f P_k^l$ where $i$ is the role of the agent in the interaction (e.g. listener, speaker, etc), $l$ is the layer number and $k$ is a unique identified within this layer- and the reverse process (${}_i^r P_k^l$).

Every forward process is a mapping ${}_i^f P_k^l : \Re^{n_i} \rightarrow \Re^{n_o}$ where $n_i$ is the number of its inputs and $n_o$ is the number of its output. The inputs of the forward BIAs representing the self (${}_c^f P_*^0$)) are connected to the outputs of the interaction perception processes and its outputs are connected to the actuators of the robot through the action integrator.The inputs of the forward ICPs at layer $l$ are connected to the activation levels of the interactive processes at layer $l-1$ through a set of $n_z^l$ delay elements. The outputs of the forward ICPs at layer $l$ are fed to the activation levels of the interactive processes at layer $l-1$. This arrangement means that processes at layer

$l$ represent an interaction protocol $n_z^l$ times slower than the processes at layer $l-1$. The activation level of the forward processes at layer $l$ are all connected to the Interaction Adaptation Manager at this layer.

Every reverse process is a mapping ${}_i^r P_k^l : \Re^{n_i+n_o} \rightarrow \Re$. The inputs of the reverse process are connected to both the inputs and outputs of its forward twin while its output which represents the expected activation level of the forward processes given the recent history of its inputs and outputs is connected to the Interactive Adaptation Manager at this layer.

In this paper it is assumed that the forward and reverse BIAs, the forward ICPs of all the roles of the interaction except the one to be learned, and their reverse counterparts are all known. It is also assumed that the forward ICPs of the role to be leaned are known up to a parameter vector ${}_i^f P_k^l$.

The Mirror Trainer in Fig. 1 is responsible of keeping the reverse interactive processes in match with their twin forward ICPs once the parameter vector of these forward ICPs are changed. This is accomplished simply by running an offline version of the forward processes while monitoring their outputs and adapting the Radial Basis Function Neural Network (RBFNN) of the reverse interactive process using this training set.

The Interaction Structure Learner is responsible of learning the number and structure of the ICPs up to a parameter vector. This component is not discussed later in this paper.

The system utilizes a set of shared variables that control the learning rate of it:

1. Age $Ag$: This variable increases monotonically (linearly in the current implementation) to a maximum of $Ag_{max}$ with the interaction time of the agent and is used to determine the reluctance to adaptation.

2. Role $Rl$: Represents the role played by the agent in the current interaction

3. Robust ${}_kRb$: Represents the average difference between the theory and the simulation for the current partner.

The learning rate of the agent is determined as: $\eta = \frac{{}_kRb}{Ag}$ where $k$ is the role to be learned.

## 3. INTERACTIVE ADAPTATION MANAGER

The interactive adaptation manager (IAM) is the heart of the proposed system and runs the interactive adaptation algorithm in all the interaction control layers starting from layer number 1. The goal of this process is to allow the agent to acquire the protocol used by a specific role agent (e.g. listener, instructor, etc) while it is doing some other role it has already learned. To achieve this goal the IAM monitors the difference between the current *theory* the agent have about what its partner (who plays the role to be learned) is intending at different levels of abstraction and the *simulation* of what it could have done if it was playing this role. The IAM then adapts the forward ICPs representing the target role to reduce this difference. In the rest of this paper we will consider two-agent interactions (e.g. lister-speaker, teacher-student etc) although extension of the technique to interactions involving more than two roles is straightforward.

The goal of the IAM is to find for every process a parameter vector ${}_i^f \hat{p}_k^l$ that minimizes the error estimate:

$$e_k^l = d\left(a_t\left({}_iP_k^l\right) - a_s\left({}_iP_k^l\right)\right)$$

where $a_t\left({}_iP_k^l\right)$ is the estimate of the activation level of process ${}_iP_k^l$ based on the theory, $a_s\left({}_iP_k^l\right)$ is the estimate of the activation level of process ${}_iP_k^l$ based on the simulation and $d\left(x, y\right)$ is a distance measure. Currently euclidian distance is used.

To achieve this goal some restrictions have to apply to the design of the ICPs. In the current implementation every ICP keeps probability distribution over the ICPs or BIAs of the immediate lower layer. This distribution is used to decide the activation level of these lower layer interactive processes as shown in Algorithm 1. What this algorithm mainly does is randomly selecting an ICP from the lower layer using the associated distribution (RAND) and then activates this ICP while deactivating the ICPs that are mutually exclusive with this ICP. In the actual implementation the inhibitory effect channels of the EICA architecture was used to do this deactivation [5].

**Algorithm 1** Interaction Control Process.

**function** INTERACTION CONTROL PROCESS(${}_ip_k^l$ ▷ Probability distribution Associated with ${}_i^fP_k^l$ , $n^{l-1}$ ▷ Number of ICPs in layer $l-1$)
  $j_{max} = RAND\left({}_ip_k^l\left(j\right)\right)$
  $a\left({}_ip_{j_max}^{l-1}\right) \leftarrow \varepsilon$
  **for** $j = 1 : n^{l-1}$ and $j \neq j_{max}$ and $MutualExeclusive\left({}_i^fP_j^{l-1}, {}_i^f Pj_{max}{}^{l-1}\right)$ **do**
    $a\left({}_ip_j^{l-1}\right) \leftarrow -\varepsilon$
  **end for**
**end function**

To generate the theory about the intention of the partner the agent runs the reverse ICPs on the outputs of the perspective taking processes representing this partner. Those ICPs output the expected activation level of their forward counterparts in the whole hierarchy. This is illustrated by the Reverse Simulated Processes block in Fig. 1. This calculation propagates bottom-up in the interaction control hierarchy resulting of $a_t\left({}_iP_k^l\right)$.

To calculate what the agent would have done using its current interaction competencies in the situation of the partner, the system runs the forward ICPs of the role taken by the partner and monitors the activation levels of those ICPs. This is depicted by the Forward Simulated Processes in Fig. 1. This calculation propagates top-down resulting of $a_s\left({}_iP^l_k\right)$.

The interactive adaptation algorithm uses these two signals to adapt the forward ICPs of the target roles and then executes the mirror trainer to adapt the reverse ICPs of the affected forward ICPs. The details of this algorithm are shown in Algorithm 2.

**Algorithm 2** Interactive Adaptation Algorithm

**function** INTERACTIVE ADAPTATION ALGORITHM($\eta$, $a_t\left({}_iP^l_{k_l}\right)$, $a_s\left({}_iP^l_{k_l}\right)$,$a_t\left({}_iP^{l+1}_{1:n^{l+1}}\right)$, $a_s\left({}_iP^{l+1}_{1:n^{l+1}}\right)$,$n^l$,$n^{l+1}$)

$n \leftarrow n+1$

$e^l_k \leftarrow d\left(a_t\left({}_iP^l_k\right) - a_s\left({}_iP^l_k\right)\right)$

**if** $e^l_k > \tau_e and {}_kRb > \tau_{Rb}$ **then**

$w_s = Ag/Ag_{max}$

$w_t = 1.0 - w_s$

${}_kRb \leftarrow \frac{(n-1)\times {}_kRb - w_s \times e^l_k}{n}$

$k^{l+1}_{\max} = \arg\max_{k^{l+1}} \left(w_s a_s\left({}_iP^{l+1}_{k^{l+1}}\right) + w_t a_t\left({}_iP^{l+1}_{k^{l+1}}\right)\right)$

${}_ip^{l+1}_{k^{l+1}_{\max}}(k_l) \leftarrow (1+\eta) \times {}_ip^{l+1}_{k^{l+1}_{\max}}(k_l)$

**for** $j = 1 : k_l$ **do**

$${}_ip^{l+1}_{k^{l+1}_{\max}}(j) \leftarrow \frac{{}_ip^{l+1}_{k^{l+1}_{\max}}(j)}{\sum_{m=1}^{n^l} {}_ip^{l+1}_{k^{l+1}_{\max}}(m)}$$

**end for**

$MirrorTrain\left({}_iP^{l+1}_{k^{l+1}_{\max}}\right)$

**end if**

**end function**

The IAM algorithm first calculates the difference between the estimations of the theory and the simulation paths and only updates the parameters of the process to be learned if this difference is above some threshold and the partner is considered robust enough to learn from him/her/it. If there is a need for adaptation:

1. The system decreases the robustness of the partner (${}_kRb$ ) based on the age ($Ag$).
2. The ID process in the process in the next layer most probably responsible of this error is calculated ($k^{l+1}_{\max}$).
3. The probability distribution of this process in the next layer (${}_iP^l_{k^{l+1}_{\max}}$) is updated to make this error less likely in the future.
4. The mirror trainer is executed to make ${}^r_iP^l_{k^{l+1}_{\max}}$ compatible with ${}^f_iP^l_{k^{l+1}_{\max}}$

## 4. EXPERIMENT

a simulation study was conducted to measure the capacity of the agent to learn how to control the gaze direction during listening while it is acting as an instructor for an agent that knows how to listen. The main focus of this study was to analyze the effectiveness of the mirror training and Interactive Adaptation to learn how to interact similarly to the agents encountered. Neither Interaction Structure Learning nor the naturalness of the resulting behavior was studied in this simulation. A simulation study rather than a real world human-agent interaction was selected because it allows us to control all the parameters of the fully designed agent, the noise levels, etc and because it can be speeded up to allow us to study more interactions (the simulations in this experiment were run 600 times faster than the real-time speed). Ten different fully designed agents were implemented that differ in the details of how they conduct instruction and how they respond to it while three agents were designed as instruct-only agents and the goal of the experiment was to study how can those agents learn listening by instructing the fully designed agents. Because the verbal content was not needed in this experiment a single 10 minutes speech was recorded and parts of it are played while instructing. The virtual environment in which the agents interacted consists of a table with six different ob-

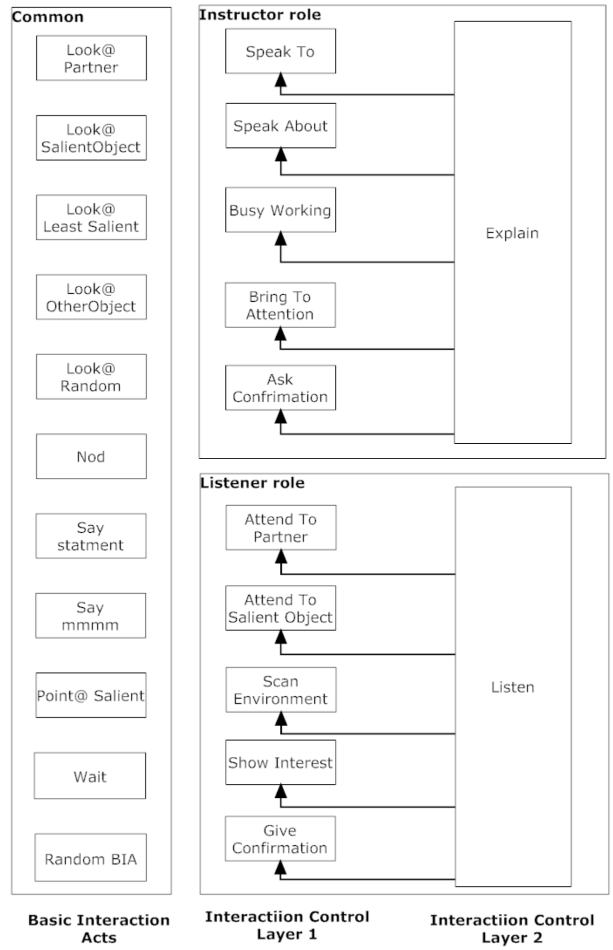


**Fig. 2**. The Processes of the agents used in the experiment.

jects and the agents were standing facing each other in the opposite directions of the table. The locations of the objects were selected randomly within the surface of the table. During the interaction, when the instructor is speaking about an object or working on it, there is a probability (7% and 10% respectively) that it will move the object. The maximum distance between the agent and the objects can be longer than its hand so the instructor has to move sometimes along its side of the table. Every agent has two arms that can be used to manipulate objects or point to them. The inputs to the agents are the 3D locations of objects along with eight position sensors attached virtually to the front and back of the heads of the agents and their right palm and index fingers (used to discover pointing). The final input channel is the speech signal of the other agent.

Fig. 2 shows the internal design of the behavioral control system used. Fully-Designed agents had all there ICPs defined based on an earlier study reported in [6] while Instruct-Only agents lack the definition of the five listening ICPs in layer 1. Learning these ICPs through the interaction was the goal of Instruct-Only agents. The Listen and Explain processes of layer 2 were implemented as probabilistic state machines and activated one ICP of layer 1 on every time step. The details of the probability distributions associated with different ICPs are reported in [5].

Every instruct-only agent instructed a randomly selected fully designed agent for a time period between 10 to 15 minutes in every interaction and conducted interactions until it finds that the difference between the theory and simulation paths is under a predefined threshold. After every instruction the instruct-only agent was doing a listening session with a randomly selected fully designed agent to measure its learning progress. The measure of the learning progress was calculated as the average RMS difference between the BIAs activation levels of the instruct-only agent and an the average fully-designed agent.

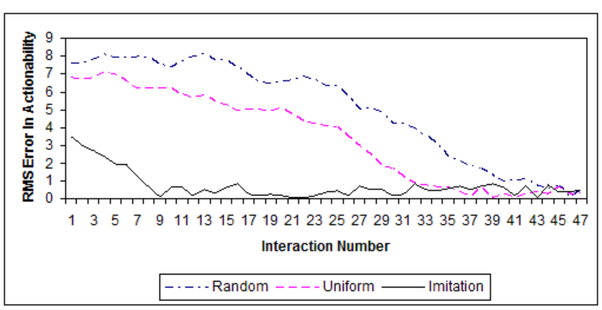


**Fig. 3**. The effect of initialization on the number of interaction needed to learn listening

Fig. 3 shows the learning progress using three different initialization settings. The error was calculated as:

$$\frac{1}{n_j}\sum_j \frac{1}{11}\sum_{i=0}^{10} a_{agent}\left(BIA_i\right) - a_{ideal}\left(BIA_i\right)$$

where $n_i$ is the number of test interactions, $BIA_i$ is the basic interaction act number $i$ and $a_x\left(P\right)$ is the actionability of the process $P$ in case of agent $x$. Three different initialization settings were used. In the first setting the listening ICPs were initialized to have a random probability distribution. In the second setting the listening ICPs were initialized to have a uniform probability distribution. In the third case imitation was used to initialize the ICPs. As the figure shows imitation boosts the speed of learning and allowed the agent to learn the required ICPs in 22% of the time needed to learn in the random initialization case and less than 27% of the time needed to learn in the uniform initialization case. The figure also shows that imitation based initialization has a much smaller initialization error (3.45).

## 5. RELATION TO CURRENT APPROACHES

Ishiguro et al. [1] proposed a robotic architecture for interactive robots based on situated modules and reactive modules in which reactive modules represent the purely reactive part of the system, and situated modules are higher levels modules programmed in a high-level language to provide specific behaviors to the robot. The situated modules are evaluated serially in an order controlled by the module controller [7].

Research in nonverbal communication in humans reveals a different picture in which multiple different processes do collaborate to realize the natural action. For example [3] showed that human spatial behavior in close encounters can be modeled with two interacting processes. It is possible in the selective framework to implement these two processes as a single behavior but this goes against the spirit of behavioral architectures that emphasizes modularity of behavior [8]. The $L_i$EICA architecture deals with this problem by allowing multiple processes to control the agent at the same time while providing a two levels action interaction mechanism that is based on actionability and intentionality that can provide coherent behavior while being able to model the complex parallel processes used to control human interactive behaviors. For details about the action integration mechanism of $L_i$EICA refer to [9].

Another more interesting difference between $L_i$EICA and the situated modules approach is the explicit representation of the nonverbal protocol at different interactive speeds in $L_i$EICA. This explicit protocol representation rather than a passive theory of mind is a unique feature of the proposed approach that can hopefully help the agent to discover how to interact and also help analyzing why the agent is interacting the way it does.

The C5m [10] architecture which is based on the work of [11] and was modified and extended by many researchers and is currently used for both synthetic character and interactive robotic research. The control system of the agent in C5m is divided into a set of motivational subsystems and every one of them is represented by a tree of actions with more abstract actions at the top and more reactive actions at the bottom. Every one of those actions is represented by a set of activation rules, pieces of code to achieve the action etc. The perceptual subsystem is represented by a tree of percepts. Each percept is a piece of code that can detect some feature in the input stream to the agent. Details of this system can be found in [10]. The common feature of $L_i$EICA and C5m is the focus on learning but this focus is presented in different ways. The learning system of C5m can be considered as an intelligent way to discover both the state and action spaces available to the agent utilizing help from a human teacher. This learning mechanism has no notion of a protocol and although effective in explicit teaching situations it is difficult to encode the multiple time scales of synchronization required for interactive agents that can be captured in the ICPs of $L_i$EICA.

In summary the main important feature of $L_i$EICA that allows it to be a better architecture for combining autonomy and interactivity is the notion of protocol and the ability to learn interaction protocols online based on the difference between the bottom-up theory and the top-down simulation paths. The ability to learn the structure of the interaction at different time scales is another advantage of $L_i$EICA although it was not discussed in details in this paper.

## 6. CONCLUSION

This paper presented a novel learning algorithm that allows an agent to learn an interaction protocol by engaging in interactions with other agents. The proposed method can be easily extended to multiparty interactions.

The proposed method allows the agent to acquire a simple theory of mind regarding the type of interactions it engages in by utilizing elements from the theory of theory and the theory of simulation.

A proof of concept experiment in a simulated environment showed the effectiveness of the approach and provided an evidence that imitation can be used as an initializing behavior to help the agent learn the interaction protocol faster.

The method as presented in this paper can only be used to learn interaction protocols and is not suitable for learning task completion behaviors. Another limitation of the proposed method is that it requires a priori knowledge of the interaction control processes up to a parameter vector. Currently we are investigating a new interaction structure learning algorithm that can hopefully overcome this limitation.

Another direction for future research is applying the proposed method to real world human robot interactions to measure its effectiveness in capturing human nonverbal behavior in explanation scenarios.

## 7. REFERENCES


[1] Hiroshi Ishiguro, Toshiyuki Kanda, Katumi Kimoto, and Toru Ishida, "A robot architecture based on situated modules," in *IEEE/RSJ Conference on Intelligent Robots and Systems 1999 (IROS 1999)*. IEEE, October 1999, vol. 3, pp. 1617–1624.

[2] C.D. Thomaz A.L. Hoffman G. Berlin M. Breazeal, C. Kidd, "Effects of nonverbal communication on efficiency and robustness in human-robot teamwork," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005)*, 2005, pp. 708–713.

[3] Michael Argyle, *Bodily Communication*, Routledge; New Ed edition, 2001.

[4] Yasser F. O. Mohammad and Toyoaki Nishida, "Intention thorugh interaction: Towards mutual intention in human-robot interactions," in *IEA/AIE 2007 conference*, June 2007, pp. 114–124.

[5] Yasser Mohammad and Toyoaki Nishida, "Toward combining autonomy and interactivity for social robots," *AI & Society*, 2008, in press.

[6] Yasser F. O. Mohammad, Taku Ohya, Tatsuya Hiramatsu, Yasuyuki Sumi, and Toyoaki Nishida, "Embodiment of knowledge into the interaction and physical domains using robots," in *International Conference on Control, Automation and Systems*, October 2007, pp. 737–744.

[7] Hiroshi Ishiguro, Tetsuo Ono, Michita Imai, Takeshi Maeda, Takayuki Kanda, and Ryohei Nakatsu, "obovie: an interactive humanoid robot," *Industrial Robot*, vol. 28, no. 6, pp. 498–504, 2001.

[8] Marc Carreras Perez, *A Proposal of a Behavior-based Control Architecture with Reinforcement Learning for an Autonomous Underwater Robot*, Ph.D. thesis, University of Girona, 2003.

[9] Yasser Mohammad and Toyoaki Nishida, "Two layers action integration for hri: Action integration with attention focusing for interactive robots," *ICINCO 2008*, May 2008, to appear.

[10] Jesse V. Gray, "Goal and action inference for helpful robots using self as simulator," M.S. thesis, School of Architecture and Planning, MIT, 2004.

[11] Bruce Mitchell Blumberg, *Old Tricks, New Dogs: Ethology and Interactive Creatures*, Ph.D. thesis, School of Architecture and Planning, MIT, 1997.