A GAME PLAYING ROBOT THAT CAN LEARN A TACTICAL KNOWLEDGE THROUGH INTERACTING WITH A HUMAN

Raafat Mahmoud, Atsushi Ueno and Shoji Tatsumi
Department of Physical Electronics & Informatics, Osaka City University, Osaka 558-8585, Japan

Keywords: Humanoid robot, Game strategy, Learning from observation, Structured interview, Long-term memory, Sensory memory, Working memory.

Abstract: We propose a new approach for teaching a humanoid-robot a task online without pre-set data provided in advance. In our approach, human acts as a collaborator and also as a teacher. The proposed approach enables the humanoid-robot to learn a task through multi-component interactive architecture. The components are designed with the respect to human methodology for learning a task through empirical interactions. For efficient performance, the components are isolated within one single API. Our approach can be divided into five main roles: perception, representation, state/knowledge-up-dating, decision making and expression. A conducted empirical experiment for the proposed approach is to be done by teaching a Fujitsu's humanoid-robot "Hoap-3" an X-O game strategy and its results are to be done and explained. Important component such as observation, structured interview, knowledge integration and decision making are described for teaching the robot the game strategy while conducting the experiment.

1 INTRODUCTION

Learning from the environment through interaction is a skill well mastered by human beings. Humans adopt their learning algorithms according to the task in which they wish to learn. For example, learning how to drive cars is different from learning how to play chess. If we are learning chess through empirical teaching class, the teacher and the collaborator is only one person. The learner plays following naive game theories at the early stages of learning procedure. In order to improve these naive theories the collaborator performs an interruption to the game events through various forms according the situation. On the other hand, the learner needs to understand the context of such interrupted situations in order to update his knowledge, and make use of it whenever needed. Therefore, the learner starts expressing his misunderstanding through various multi-modals interactions. The interaction between the learner and his teacher improves the learner understanding level about these situations. In these cases, the learner's brain processes these situations to store certain information about these situations, which improves the learner's naive theories of the game. Theoretically based on many cognitive researches, human learning is assumed to be "storage of automated schema in long-term memory of human brain" (Sweller, J 2006). Schema is chunks of multiple individual units of memory that are linked into a system of understanding (Bransford, J., Brown, A., 2001). However, the learner's brain performs many processes to the input data at different places, such as short-term memory and working memory (Baddeley, A. D 1996). Then a certain extracted data is stored at its long-term memory. The short-term memory is assumed as the place where experiencing any aspect of the world. Working memory is a place where thinking gets done. It is actually more brain function than a location. The working memory is dual coded with a buffer for storage of verbal/text elements, and a second buffer for visual/spatial elements (Marois, R. 2005). The main function of Long-term is storing the learned data as a schema to make use of it when ever needed, without the need of learning the subject from the first steps again. The main phases of the learner's brain in learning such a task are observation of the task sequences, interviewing about what we do not understand, recording the concluded data from observation and interviewing, process tracking and making use of the recorded data to support a hypothetical scenarios for making decisions. In this paper, we describe an architecture through which a human teacher teaches a humanoid-robot "Hoap-3" the game strategy through an interaction algorithm which humans follow in order

Mahmoud R., Ueno A. and Tatsumi S..
A GAME PLAYING ROBOT THAT CAN LEARN A TACTICAL KNOWLEDGE THROUGH INTERACTING WITH A HUMAN.
DOI: 10.5220/0003182406090616
In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence* (ICAART-2011), pages 609-616
ISBN: 978-989-8425-40-9

to learn a task for the first time. Many other architectures for teaching a robot by demonstration were introduced (Kuniyoshi. Y., M. Inaba. M, and Inoue. H. 1994)(Voyles. R and Khosla P. 1998). However such approaches use demonstrations in order to optimize a predefined goal, and also the interactive behaviours follow human-machine (Reeves, B. & Nass, C. 1996) interaction, but does not follow human-human interaction, which come out of the strict paradigm that robots are following. A tutelage and socially guided approach for teaching the humanoid robot "Leo" a task (Lockerd A., Breazeal C. 2004) was proposed, where machine learning problem is framed into collaborative dialogue between the human teacher and the robot learner, however every task has a specific single goal. In our approach making decisions is based on accumulative learning that "Hoap-3" gains while interaction, additionally adaptive selections behaviour for each new situation in order to achieve individual goals based on the accumulative learning information. As an architecture about learning and interacting in human-robot domain and task learning through imitation and human-robot interaction (Nicolescu M. N., Mataric M. J 2001), a behaviour based (Barry Brian Werger, 2000) interactive architecture applied to a Pioneer 2-DX mobile robot is proposed. In these approaches the behaviours are mainly built from two components, abstract behaviour and primitive behaviours. However these two architectures are not suitable and flexible enough to be applied for teaching a robot various tasks through interaction. Also this method in various forms has been applied to robot-learning for different single-task such as hexapod walking (Maes P. and brooks R. A. 1990), and box-pushing (Mahadevan S. and Connell J. 1991). Many other single task navigation and human-robot instructive navigation (Lauria S., Bugmann G., 2002) have been proposed. In our proposed architecture there is no data provided in advance, and the goals of a task are being taught while interactions.

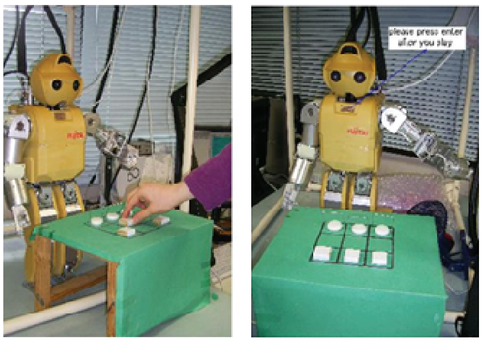


Figure 1: The robot interacts with its human teacher.

Moreover, the interactive behaviours are resembled to those of human's behaviours while learning. In this paper, the main features of our architecture and the developed behaviours are explained at the following sections. Following this section, the internal system structure is explained. Then decision making process is explained. At the last two sections, testing our architecture and results from an experiment are explained. This is followed by discussions about our architecture.

2 OUR ARCHITECTURE

In our approach we use an upper torso of a humanoid robot "Hoap-3", which has a total 28 degree of freedom (DOFs), 6 flexibility degrees in each arm, and other 6 flexibility in each leg, 3 flexibility degrees in the head, one degree in the body (see figure 1).

In order to provide an interactive learning behaviour, the architecture must be flexible. This improves the internal processing strategy between the architecture components, which enables the robot to recognize and identify its environment correctly, which in return, improves the efficiency of mapping between the robot expression components. This flexible system provides perfect interactive behaviours in response to its environment changes. To achieve such an aim, we designate architecture which composed of multi-components within a single API root layer (see figure 2).

Our architecture has components requiring information from the system, such as environment handling, knowledge updating and expressions, and other components which provide information to the system, such as streaming information from the environment through a vision sensor. In addition, it has intermediaries components that provide the necessary information within the system. In our design we isolated these components from each other. A proper combination of these components can perform fair specialized behaviours. The next subsection will describe and explain such interactive behaviours.

2.1 Interactive Expressions

The expressions performed by the robot must be influenced by the task at which the robot is interacting with, and of-course in addition to the internal final information that resulted from processing the task events. However, behaviours performed by the robot are mainly low level-behaviours.

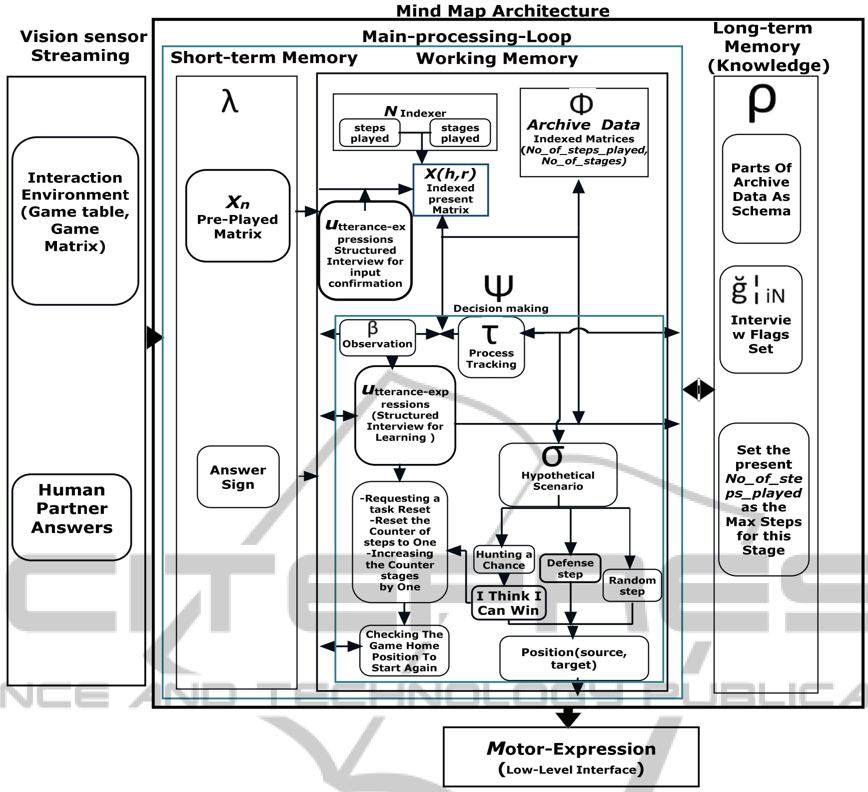


Figure 2: Hoap-3 mind-map.

```
Main_Loop() // Where the work gets done
{
Do-Processing (): feature detection-
observation,confirmation/representation,
state/knowledge, decision-making.etc.
}
-------------------------
Void Vision_sensor_component
{
While: true
Stream into the main processing loop
}
------------------------
Motor_expression(source, target)
{
While: true
Get-from-main-process-loop (source,
target)

Set trajectory
Perform-low-level-interface
}
```

Figure 3: Main components of our architecture.

The low-level robotic behaviours mainly include processing the streamed signals from the sensors, and performing low level interface to the robot actuators. A proper combination of such individual low-level behaviours enables the robot to produce a higher level behaviour. The robot expression may be *motor-expression M* or *utterance-expressions U.*

$$M = \{ m_1, m_2, m_3, \dots, m_i \} \quad U = \{ u_1, u_2, u_3, \dots\dots, u_i \} \tag{1}$$

Motor-expression is a component that provides the mapping from high-level commands to low-level motor commands that are physically realized as high-level behaviours while executing a task. It consists of collection of trajectory motor angle algorithms m_i at any step of the task in low-level parameters that provide a task execution to be done.

The utterance-expression is a collection of individual spoken words u_i at any step of the task. A proper combination of the individual words u_i produces high-level interactive behaviours while interacting with the robot. In our architecture, since the other components have the knowledge of contextual state information, the developer responsible for utterance-expressions does not need to worry about this contextual information.

In the approach of learning and playing a game such as X-O game, the robot must have the capability of moving an object from one place to another. The source and the target of an object are specified by decision-making process. However, a source of an object at a situation may become a target at another situation. Therefore, and in order to avoid the duplicating the software code and inverse kinematics calculations, a software plug-in is to override the system and handle such a conflict by controlling the robot gripper. Mapping and selecting the *motor-expression M* or *utterance-expressions U* are made by the *Decision-making* (will be explained later) based on the information stored at the long-term memory; also the conclusion depends on the environment events computed by the perception component. In the next section we will describe the internal components strategy in processing the information while teaching the robot a task.

3 SYSTEM ARCHITECTURE

In this section, to show a theoretical instantiation of the architecture and procedure of interacting with a robot, X-O game developed between a human partner and "Hoap-3" is referred. X-O game consists of 3*3 square board, and six game pieces. Three for a human partner (*h* square parts and indexed as 2 while processing) and the other three are for the robot (*r* round parts and indexed as 15 while processing). Human plays first, and then the robot plays (see figure 1). Winning is achieved when one of the players assembles a complete line (row or column). Human partner should play only one of his own game pieces at his game turn, and then prompts the robot to play, and it is the same for the robot. It only should play one of its game pieces, and then prompts its human partner to play. We divided the X-O game into stages, when one of the players wins, or the robot learns a new idea; a new stage starts from the home position all over again. While playing, the robot indexes every step played by the indexer (see figure 2); and records it as a variable named *no_of_steps_played*, also indexes every stage and records it as a variable named *no_of_stage*. A new stage starts if *no_of_steps_played* is reset to one. *no_of_stage* is increased by one while the game going. In the approach of teaching a humanoid robot the game strategy, we aim to teach the robot a high level behaviour performed by a human partner.

The final information from low-level signals of the vision sensors results in a matrix *X(h,r)* that includes the number of the human game pieces *h,* and the robot game pieces *r* at every game step as shown in figure 4.

2	2	2
0	0	0
15	15	15

Figure 4: The final matrix resulted from vision sensors processing (home position matrix).

While playing, this matrix *X* is stored as archive data Φ as in equation (3)(see figure 2).

$$\Phi = \{X_{(1,1)}, X_{(2,1)}, X_{(3,1)}, \ldots, X_N\}, \qquad (2)$$

where *N* is the index number (*no_of_steps_played, no_of_stage*). Another place to store matrix X is the sensory memory as in human brain, however, in the sensory memory register λ, we only store a single piece of data X and replace every game step, as explained in equation (3).

$$\lambda = X_n \qquad (3)$$

Where X is piece of data that denotes the game matrix, however *n* denotes to the index coordinate (*no_of_steps_played* - 1*, no_of_stage*) of the game step.

At every game step, human teacher performs action *δ*. These actions are general actions or have a specific purpose ğ as equation 4 shows.

$$\delta = \{ \delta \,|\, \breve{g}_{1|N}, \delta \,|\, \breve{g}_{2|N}, \delta \,|\, \breve{g}_{3|N}, \ldots, \delta \,|\, \breve{g}_{i|N} \} \qquad (4)$$

The preceding equation δ denotes the action made by human teacher along task playing and teaching (general action), and $\delta|\breve{g}$ denotes to the actions such that a specific purpose $\breve{g}$ is achieved at game step *N*. In our task we have specific goals, such as teaching the robot a winning or defence movements (*is_winning* or *is_defence)* for the human teacher or for the robot (*my_wining, my_defence*) as explained in the next equation(5) which denotes that, for every special action *δ* a specific purposes $\breve{g}_N$ at *N* index.

$$\breve{g}\,|_{iN} = \{is\_winning,\ my\_wining,\ is\_defence\ my\_defence\} \qquad (5)$$

3.1 Observation of Human Behaviour

In many proposed approaches (Brian S, Gonzalez J. 2008), templates were provided in advance in order to assist the system to recognize the context an action. Also in another proposed approach (Mahnmoud, R. A., Ueno. A., Tatsumi, S., 2008), a knowledge data are provided in advance. However, in our architecture we extract the individual low-level behaviour context which leads the robot to the high-level behaviour learning by applying the following algorithm.

Starting from low-level processing, at which the robot is able to identify the game pieces coordinates according the 2-D camera frame and obtain the game matrix *X(h,r)*. This contains the three pieces of the human teacher *h* and the other three pieces for the robot *r*. The robot should have the ability to recognize the high-level behaviour performed by human teacher. To do so, the observation component in our architecture performs the following processes. First the human game pieces are replaced with zeros, which results in a matrix $X|_{N\text{- Hoap-3''-Part}}$ that includes only the robot game pieces as follows;

$$X|_{N\text{- Hoap-3"-Part}}\,(h_{|=0}, r)$$

Then the robot's game pieces are replaced ones in which X becomes a logical matrix $X|_{N\text{-"Hoap-3"-Part}}$, and includes only the spatial coordinates of robot's game pieces at *N* step as follows;

$$X|_{N\text{-"Hoap-3"-Part}}\,(h_{|=0}, r_{|=1})$$

On the other hand the same processes are performed to the same matrix *X(h,r)* but for the teacher's game pieces, and produces a logical matrix $X|_{N\text{-Teacher-Part}}$ which includes the spatial coordinates of the teacher's game pieces at the same *N,* as follows;

$$X|_{N\text{-Teacher-Part}}\,(h_{|=1}, r_{|=0})$$

Also the same processes are being performed to the data $\lambda = X_n$ in equation (3) resulting two matrixes, the first one is logical matrix $X_{|n\text{-"Hoap-3"-Part}}$, includes only the spatial coordinates of robot's game pieces at *n* step, and another logical matrix $X|_{n\text{-Teacher-Part}}$ and includes only the spatial coordinates of teacher's game pieces at the same step *n* as follows;

$$X|_{n\text{-"Hoap-3"-Part}}\,(h|_{=0}, r|_{=1})$$

and

$$X|_{n\text{-Teacher-Part}}\,(h|_{=1}, r|_{=0})$$

In order to obtain the context of the low level behaviour performed to the task is compare the both the data in X_N and X_n in a special manner using Ex-or logic gate as in the syntax followed in the two equations (6)(7);

$$Đ|_{\text{Teacher-Part}} = (X|_{N\text{-Teacher-Part}}(h|_{=1}, r|_{=0}) \text{ EX-OR } X|_{n\text{-Teacher-Part}}\,(h|_{=1}, r|_{=0})) \tag{6}$$

and

$$Đ|_{\text{"Hoap-3"-Part}} = (X|_{N\text{-"Hoap-3"-Part}}\,(h|_{=0}, r|_{=1}) \text{ EX-OR } X|_{n\text{-"Hoap-3"-Part}}\,(h|_{=0}, r|_{=1})) \tag{7}$$

The resultant data of this procedure is called an observation data β as shown in equation (8);

$$\beta = < Đ|_{\text{Teacher-Part}}\,,\ Đ|_{\text{"Hoap-3"-Part}} > \tag{8}$$

The resultant information from the observation is one of three cases directives statuses, status one < status=No pieces have been moved >, if

$$\beta = <0,\ 0>$$

Status two indicates <status= the robot game piece has been moved>, if

$$\beta = <0,\ 1>$$

And finally status three <status="User-Teacher" piece has been moved>, if

$$\beta = <1,\ 0>$$

In addition to this, the observation component at our architecture is able to identify the spatial coordinates of the game piece which has been moved. These bundles of data are submitted to the *Decision-making* process as will be explained at the next section, in which the appropriate action is to be selected.

4 DECISION MAKING PROCEDURES

During the interaction procedure, the human teacher sometimes plays random steps. In this case the concluded information from the observation process β, are sent to the *Decision-making* process ψ as.

$$\Psi\,(X, \Phi, \beta, \delta, \tau, \sigma, \rho)_{|N} = \sum\nolimits_{q=1}^{i} B \tag{9}$$

In the proceeding equation, the Decision-making main frame Ψ produces a number *q* for an individual behaviours B orchestrated by the robot at any step *N* while interacting with the human teacher. The behaviour B is a combination of the *motor-expressions M* and/or *utterance-expressions U.*

$$B = \ < M, U >$$

In order to orchestrate a suitable behaviour in response to the interaction situation, the decision-making process Ψ, subscribes to the information resulted from the observation component β, and also the decision-making process subscribes to the archive Φ by performing a process tracking procedure τ shown in equation (10);

$$\tau = X_N \cap \Phi \tag{10}$$

This resulted information from the process tracking τ provides Ψ the necessary information in order to perform hypothetical scenario σ, which is the resultant data from the union of process tracking information τ and knowledge data ρ as shown in equation (11). This enables the robot to predict and decide the new step of the task which as follows;

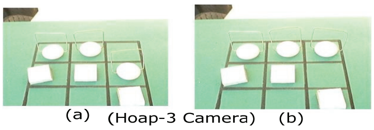
$$\sigma = \tau \cup \rho \tag{11}$$

The knowledge ρ which the robot obtains through interacting with its human teacher which as follows;

$$\rho = \{X|_m \,|\, \delta_{|1},\ X|_m \,|\, \delta_{|2},\ X|_m \,|\, \delta_{|3}, .., X|_m \,|\, \delta_{|i}\} \tag{12}$$

Where X is the matrix resulted from the interacting with the human teacher, at the *ith* situation which the human teacher teaches the robot a specific action δ. This matrix is stored at the long-term memory, *m* is the index as in ((*no_of_steps_played, no_of_stage, max_no_of_steps_played*) where *max_no_of_steps_played* is the end step played for the same stage index *no_of_stage.*

As explained the robot is able to interact with the human while learning the task through different expressions. Let us consider the following interactive situations that occurred while teaching the robot the X-O game.



Hoap-3: Why did you move my piece not your piece, is this a wining chance?
Human partner: YES
Hoap-3: is this my wining?.
Human partner: YES
Hoap-3: That is very good, please reset the game set to start over again and then press enter after you play.

Figure 5: Teaching "Hoap-3" how to achieve winning.

5 TESTING AND EVALUTING OUR ARCHITECTURE

In order to evaluate the proposed architecture, we have performed an experiment in which various interactive situations have been taken place, and among these situations, a situation at which a winning chance is available for the robot as in figure (5-a). However as there is no any data provided in advance, the robot will not be able to recognize it. Human teacher, at his playing turn, performs an interrupting step by moving the robot's game piece instead of his game pieces to set the winning row as in figure (5-b), then prompts the robot to play. The robot applies low-level identification, starting from analysing the data streams from the vision sensors, and obtains the resultant matrix $X\ (h,r)$, which is stored as a archived data Φ (see equation 2). On the other hand a single piece of data λ (see equation 3) (see figure 2) which in our present situation is the matrix in figure 5-a. After applying the observation algorithm in equations (6) and (7) which leads to higher level observation β at equation (8), the following β is obtained;

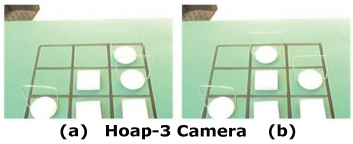
$$\beta = <0, 1>$$

This information is submitted to the Decision-making procedure Ψ, which orchestrates number q of individual behaviour B such as moving the robot's upper-torso in addition to its arm through the arm motor-expressions *motor-expressions M,* in addition to this, the Ψ orchestrates an *utterance-expressions U* as in figure 2.

The utterance-expression provides the necessary information as

$$\breve{g}_N=\{\ \breve{g}_N\big|_{is\_winning}=1,\ \breve{g}_N\big|_{my\_wining}=1,\ \breve{g}_N\big|_{is\_defence}=0,\ \breve{g}_N\big|_{my\_defence}=0\},$$

which purify the purpose of the human-teacher action δ.



Hoap-3: Why did you move my piece not your piece?, is this a wining chance?.
Human Teacher: No.
Hoap-3: You mean this is a defence step, right?, Is this my defence ?.
Human Teacher: YES.
Hoap-3: That is very good, please reset the game set to start over again and press enter after you play.

Figure 6: Teaching "Hoap-3" how to make a defence step.

As the structured interview shows, the robot asks the human teacher to reset the game set in order to restart a new stage.

On the other hand the knowledge ρ of the robot must be updated. Therefore the structured interview result $\breve{g}_N$ and the two matrixes as in figure (5-a, 5-b), are stored at a different register as a long term memory as a knowledge data ρ at index m (see figure 2). Noting that the index N is turned into m which holds the index N in addition to storing *no_of_steps_played* as *max_no_of_steps_played* which is useful for the robot whenever using the knowledge ρ data for making a decision.

Another situation that is a chance to teach the robot how make a defence step is available as shown in figure (6). For the pesent situition the same procces starting from $X\ (h,r)$, until updating the knowledge ρ takes place. The only different is the structured interview resultant data $\breve{g}_{N..}$ Which is follows;

$$\breve{g}_{NX}=\{\ \breve{g}_N\big|_{is\_winning}=0,\ \breve{g}_N\big|_{my\_wining}=0,\ \breve{g}_N\big|_{is\_defence}=1,\ \breve{g}_N\big|_{my\_defence}=1\}$$

This leads to inform the robot the high level context of the teacher's action δ. Also an updating the knowledge data ρ is being performed.

In a different situation, at which the human teacher aims to teach the robot the form of his winning as shown in figure (7-a). Human plays and gets the available winning chance as in figure (7-b), and then he prompts the robot to start playing. Now we should start a new playing stage due to the wining that was achieved for human partner.

As there is no data provided in advance, the robot will not recognize it and starts to play randomly and may be as in figure (7-c), and then it prompts its human partner to play. In order to teach "Hoap-3" how human wining is achieved, human partner will not move any of the game pieces then it prompts the robot to play. In this situation high-level observation β and results as follows;

$$\beta = <0, 0>$$

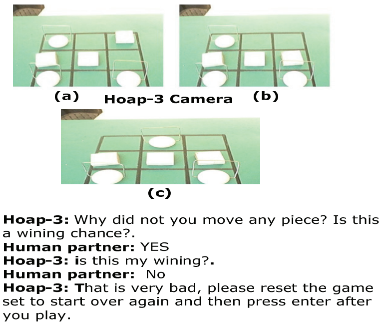


Figure 7: Teaching "Hoap-3" how human winning is achieved.

This means there is no any of the game pieces have been moved. This data is submitted to decision making main frame Ψ which orchestrates a new structure interview based on the real-time interaction as figure 7 shows. Also the same procedure is followed by the robot. However the resultant data from the structured interview $\breve{g}_N$ is different for the previous two situations, which as follows;

$$\breve{g}_{NX}=\{\ \breve{g}_N \mid is\_winning=1,\ \ \breve{g}_N \mid my\_wining=0,\ \breve{g}_N \mid is\_defence=0,\ \breve{g}_N \mid my\_defence=0\}$$

Also another different in this situation is that the data XI_m that submitted to knowledge updating ρ includes the matrix figure 5-a.

During the interaction procedure, the human teacher sometimes plays random steps. In this case the observation data is obtained as follows;

$$\beta =< 1{,}0 >$$

This informs "Hoap-3" that the movement made by its human teacher is a regular step. In this case decision making process Ψ performs a different procedure from the previously explained situation.

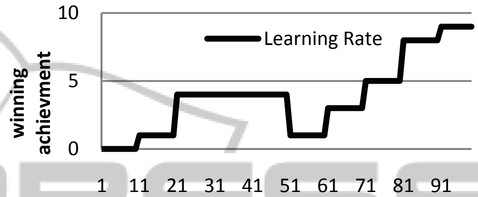
The first procedure is performing process tracking τ by matching the present matrix X_N with the archive data Φ as explained in equation 10 (see figure 2) if τ = < empty> the robot plays randomly (see figure 2). However, if τ ≠ < empty> the robot unites the resultant data from τ with the knowledge data ρ.

The knowledge data ρ includes the high level context of every interactive action $\delta \mid \breve{g}_N$ made by its teacher. Based on this union, the robot performs a hypothetical scenario σ in order to make a rational choice. However, if the process tracking is τ > 1 then the hypothetical scenario's σ main priority is given to choose knowledge as follows;

$$\breve{g}_{NX}=\{\ \breve{g}_N \mid is\_winning=1,\ \breve{g}_N \mid my\_wining=1,\ \breve{g}_N \mid is\_defence=0,\ \breve{g}_N \mid my\_defence=0\}$$

Table 1: Statistics of teaching experiment.

Order of the tenth sample space	"Hoap-3" Winning Achievement	"Hoap-3" Interviewing its Human-teacher
First 10th sample space	0	10
Second 10th sample space	1	9
Third 10th sample space	4	6
Fourth 10th sample space	4	6
Fifth 10th sample space	4	6
Sixth 10th sample space	1	9
Seventh 10th sample space	3	7
Eighth 10th sample space	5	5
Ninth 10th sample space	8	2
Tenth 10th sample space	9	1



If the hypothetical scenario σ > 1, then the robot's final decision Ψ is by choosing an action resembles the stage which has the minimum difference between *max_no_of_steps_played* and *no_of_steps_played* of the X_N at which its main priority is achieved.

$$\Delta\mid_{min}=max\_no\_of\_steps\_played - no\_of\_steps\_played$$

The second priority is given to

$$\breve{g}_{NX}=\{\ \breve{g}_N \mid is\_winning=0,\ \ \breve{g}_N \mid my\_wining=0,\ \breve{g}_N \mid is\_defence=1,\ \breve{g}_N \mid my\_defence=1\}$$

Also if the hypothetical scenario σ > 1 "Hoap-3" final decision Ψ is by choosing an action resembles the stage which has the minimum difference between *max_no_of_steps_played* and *no_of_steps_played* of the X_N.

From these combinations, the robot is able to select only rational choice, then the robot says as follow:

Hoap-3: *I think I can win.*

Among the individual B (see equation 9) expression which the robot performs various motor expressions are made such as upper-torso, hip movements, head movements, and arms movement. These expressions improve and imply the human-human behaviour.

6 RESULTS

In order to show the efficiency of our proposed architecture, we performed an experimental test composed of 100 stages and its sample space is as shown in Table 1. New stage occurs if the robot learns new idea about the winning or defence for itself or for the human. Also if a winning case of the taught ones to the robot is performed by the robot itself. The results at the table are indicated at the graph, shows that the rate of winning achieved by the robot is increased gradually, which indicates that robot learning level is increased by the increasing the number of interactive stages. This is a clue for improving robot knowledge of the game strategy.

7 DISCUSSION

We will now reflect some design issues on our robot architecture from two perspectives: component design and communication of information between components.

7.1 Information Generation

An important requirement is the need of building an approach that is able to generate new valuable information to be based and resulted from the available information. For example, in the X-O game, observation component is able to detect the spatial positions of the moved game piece with respect to the camera frame in terms of 2-D. This coordinates information is processed by position component and transformed into 3-D, and transferred to knowledge-updating, allowing "Hoap-3" to use when executing knowledge based decisions.

7.2 Information Flow

In order to improve the overall system responsiveness, we have found that one-to-many information flow structure is very useful. Where, the information is produced by one component and published to the system, where, other components process this information for their own purposes. For example, during the X-O game, the human partner performs interruptive movements to the game; observation component detects these interruptive events. The resultant detected information is published to the rest of the system. Simultaneously, the published information is handled by other component. The decision-making process uses this information in order to decide the proper choice of wording of the structured interviews. Meanwhile, the detected information in addition to the resultant interviewing flags are used to update "Hoap-3" knowledge.

REFERENCES

Sweller, J., 2006. Visualization and instructional design. *In3rd Australasian conference on Interactive entertainment*, Vol, 207,pp 91-95.

Bransford, J., Brown, A., & Cocking, R, 2001. *How people learn. Brain, Mind, Experience, and School*. Expanded version. National Academy Press: Washington, DC. p. 33.

Baddeley, A. D., 1996. *Human Memory: Theory and Practice.* Hove: Psychology Press.

Marois, R. 2005. Two-timing attention. Nature Neuroscience. In *Nature Neuroscience*. pp 1285-1286.

Kuniyoshi. Y., Inaba M., and Inoue H. 1994. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. In *IEEE Transactions on Robotics and Automation,* vol 10 pp.:799–822.

Voyles. R and Khosla P., 1998. A multi-agent system for programming robotic agents by human demonstration. In *Proceedings of AI and Manufacturing Research Planning Workshop.*

Reeves, B. & Nass, C. 1996. *The media equation—how people treat computers, television, and new media like real people and places*. Cambridge, UK: Cambridge University Press.

Lockerd A., Breazeal C. 2004 Tutelage and socially guided robot learning". In *Proceedings. International Conference on Intelligent Robots and Systems, IEEE/RSJ* Vol. 4, pp. 3475 – 3480..

Nicolescu M. N., Mataric M. J., 2001. Learning and interacting in human-robot domains*"* In *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* Vol. 31 No.5. ,pp. 419 – 430.

Barry Brian Werger, 2000. Ayllu: Distributed port-arbitrated behaviour-based control. *In Proc., The 5th Intl. Symp. On Distributed Autonomous Robotic Systems*, Knoxville, TN, pp. 25–34

Maes P. and Brooks R. A. 1990. Learning to coordinate behaviours. In *Pros AAAI, Boston, MA,* pp.796-802.

Mahadevan S. and Connell J., 1991. Scaling reinforcement learning to robotics by exploiting the subsumption architecture. In *Proc. Eighth Int. Workshop Machine Learning,* pp.328-337.

Lauria S., Bugmann G., 2002 Mobile robot programming using natural language*. Robotics and Autonomous Systems,* 38(3-4):171–181.

Brian S, Gonzalez J. 2008. Discovery of High-Level Behaviour from Observation of Human Performance in a Strategic Game. In *Systems and Humans, IEEE Transactions on* Vol. 38 No.3, pp. 855 – 874.

Mahnmoud, R. A., Ueno, A., Tatsumi S., 2008, A game Playing robot that can learn from experience. In *HSI'08 on Human System Interactions* pp. 440-445.