International Journal of Mechanical Engineering and Robotics Research

ISSN 2278 – 0149 www.ijmerr.com
Vol. 1, No. 2, July 2012


*Research Paper*

# TOWARD PERMISSIVE TEACHING THROUGH INTERACTIVE ARCHITECTURE

**Raafat Mahmoud[1]*, Atsushi Ueno[1] and Shoji Tatsumi[1]**

**Corresponding Author:* ***Raafat Mahmoud,*** ✉ *rafattheone@gmail.com*

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. Important components in our approach such as, decision making, process tracking, observation and knowledge data are described for teaching the robot a task. 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.

***Keywords:*** Interactive teaching, Learning from observation, Structured interview

## INTRODUCTION

The learning from interaction formulation has been considered in a number of studies (Fikes and Nilsson, 1971; and Penberthy and Weld, 1992) in order to be applied at different practical applications. However the past introduced issues were mainly concentrated on learning classical tasks; and its planning formulation typically assume that the learning phase was done under fully observable environment and not with a supervisor; the assumption of fully observable environment is mainly for learning a single goal from an interruptive action made by the teacher. In other words, the learning phase was not able to identify the exact purpose of the learned actions if there were various goals which are required to be learned for a single interruptive action while learning the task. Therefore, in the learning phase it should only be provided with a single specific action to be learned. Also the planning domain was limited to reach a single goal, and it could not handle extended goals such as tactical planning that handle situations

[1] Department of Physical Electronics & Informatics, Osaka City University, 3-3-138 Sugimoto Sumiyoshi-ku, Osaka-shi, 558-8585 Japan. E-mail: ueno@info.eng.osaka-cu.ac.jp; and tatsumi@info.eng.osaka-cu.ac.jp

that always change during the task execution. The union of learning extended goal tasks and supervised interaction has been rarely studied due to its hardness. In this paper, we will propose a novel technique for learning an extended goal task through interaction. Given a task to be composed of some states, for every state there is bundle of information for representing the present state. Through these states different goals can be achieved. It is required to teach the humanoid robot these types of goals through performing interruptive actions made by the teacher. Also it is required that the humanoid robot is to be able to identify the contextual information of the situation and classify the types of the interruptive teaching actions which are being learned. For this problem, our solution consists of the following three phases: 1) collaborate action phase, i.e., observing the interruptive teaching action and classifying their teaching goals; winning goals or defence goals, and whose goals they are; 2) knowledge formalization phase, i.e., extracting a specific data block in order to use it to up-data the knowledge, organizing these blocks of data and storing them as a schema in the long-term memory; and 3) decision making phase, i.e., constructing hypothetical scenarios by building a virtual structure for the present situation searching for the optimum solution and a method to achieve the best result by following a greedy adoptable thinking in which a safety requirement and a reachability requirement is adopted as the main achievement. The learning phase in our approach first uses adoptive active learning we designed in order to find the classification of the goals being learned. Notice that the passive learning learns a target from given sets of positive samples and negative samples. While active learning is supervised learning which converges to the target by asking queries to a teacher who should be able to answer correctly, this procedure takes place during the collaborate action phase in our proposed approach, in which the robot needs interactive teaching actions from its teacher. The robot asks queries to a teacher about these actions to obtain the main purpose of these actions. The teacher in our approach provides some examples of different goals through interruptive teaching actions while the task is on, and then the humanoid robot should be able to observe and identify these interruptive actions and their types and could start interviewing the teacher a structured interview about these actions. The structured interview is a set of various query sequence depending on the situation itself; the humanoid robot keeps on interviewing the teacher until it is able to understand the goal of the interruptive action. Therefore we need a teacher who can answer any structured interview about the goals of the task.

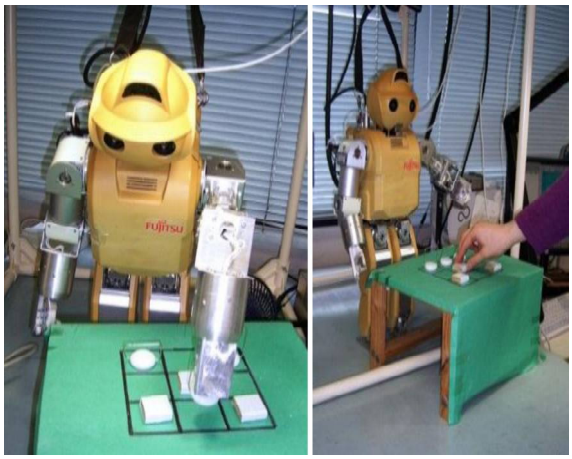
Many other architectures for teaching a robot by demonstration were introduced (Kuniyoshi *et al.*, 1994; and Voyles and Khosla, 1998). However such approaches used demonstrations in order to optimize a predefined goal, and also the interactive behaviours followed human-machine (Reeves and Nass, 1996) interaction, but did not follow human-human interaction, which came out of the strict paradigm that robots were following. A tutelage and socially guided approach for teaching the humanoid robot “Leo” a task (Lockerd and Breazeal, 2004) was proposed, where machine learning problem was framed

into collaborative dialogue between the human teacher and the robot learner, however every task had 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 and Mataric, 2001), a behaviour based (Barry, 2000) interactive architecture applied to a Pioneer 2-DX mobile robot was proposed. In these approaches the behaviours were 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 and Brooks, 1990), and box-pushing (Mahadevan and Connell, 1991). Many other single task navigation and human-robot instructive navigation (Lauria and Bugmann, 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. 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.

## OUR ARCHITECTURE

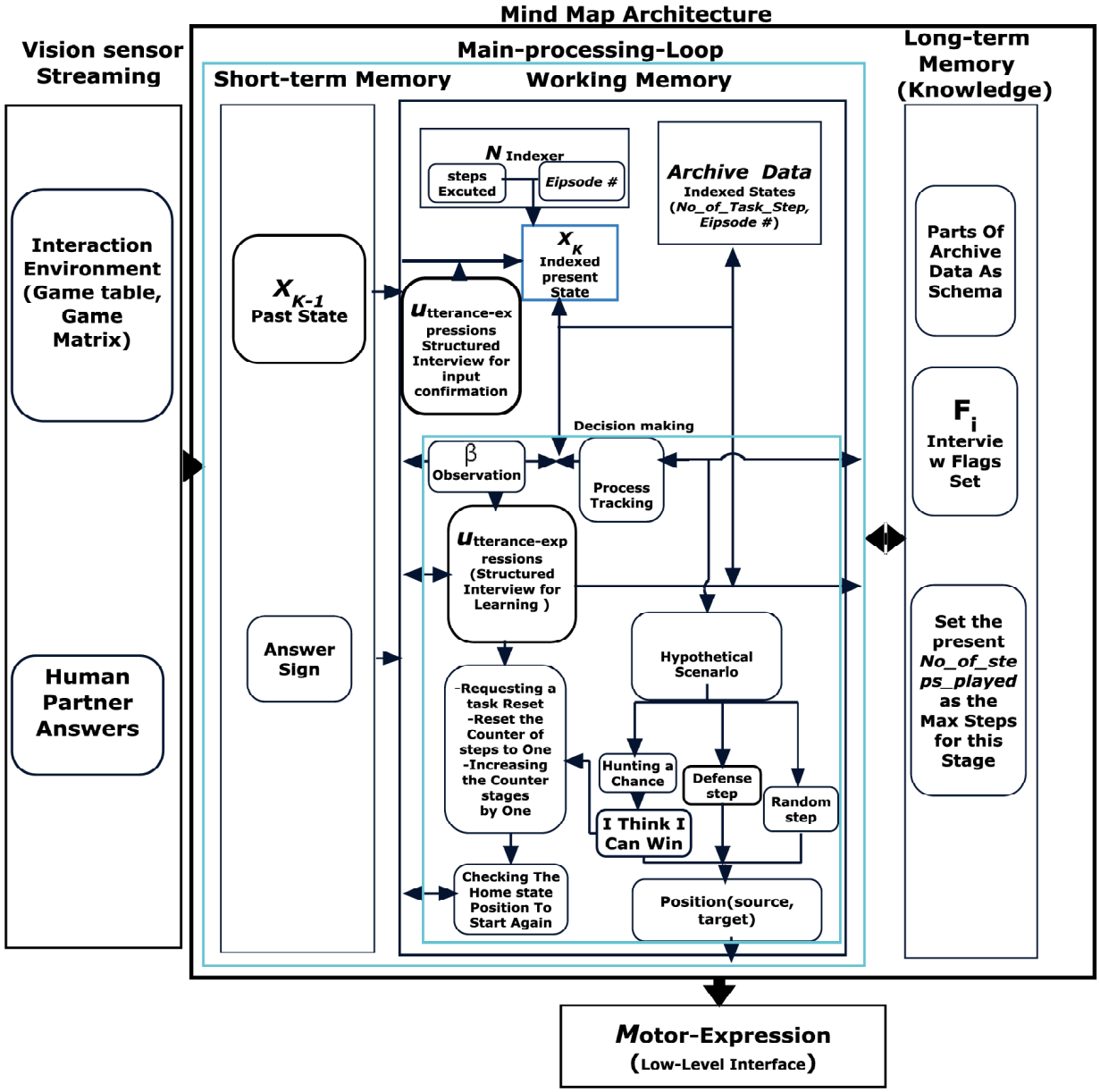
In our approach we use an upper torso of a humanoid robot "Hoap-3", which has a total 28 degree of freedom (DOFs), 6 flexibility degrees in each arm, and other 6 flexibility in each leg, 3 flexibility degrees in the head, one degree in the body (Figure 1).

**Figure 1: Humanoid Robot Interacting with its Teacher**



In order to provide an interactive learning behaviour, the architecture must be flexible in order to improve the internal processing strategy between the architecture components, which enables the robot to recognize and identify its environment properly in addition to improving the efficiency of mapping between the robot's internal expressions components. Such flexible system provides proper interactive behaviors in response to its environment. For achieving this flexibility of an architecture we designate an architecture which is composed of multi-components within a single API root layer (Figure 2). Our architecture contains components that require

**Figure 2: Components of Our Proposed Architecture**




information, such as environment handling, knowledge updating and expressions, and other components which provide information to the system architecture, such as streaming information from the environment through a vision sensor. In addition, it has intermediary components that provide the necessary information within the system architecture. As Figure 2 shows, our system architecture is divided into three main segments; the first segment is the short-term memory $\lambda$ at which an information $X_{K-1}$ about a past state of the task sequences and the answers to the interview queries are received, according to the directions of the arrows in Figure 2 these information are handled by to the second

segment which is working memory, while the information $X_K$ about the present state of the task is corrected through utterance expressions $U$ before being indexed by the step indexer $N$, and then stored as archive data $\Phi$ (see the arrow direction) which classified as intermediaries component that accessed by the decision making process $\Psi$. The observation component $\beta$ is a component that takes place between the information $X_{K-1}$ about the past state at the sensory memory and the information $X_K$ about the present state of the learned task at the working memory. The observation component $\beta$ is also classified as intermediary component that informs the decision making process $\Psi$ with the appropriate information about the process sequence status of the task and takes place during the collaborating action phase. On the other hand, the hypothetical scenario process $\sigma$ in the decision making process $\Psi$ acquires knowledge information $\rho$ from the third segment which is the long-term memory after performing process tracking $\tau$ in order to make a rational decision. The concluded decision obtained from the hypothetical scenario process is to be sent to the motor expressions component to be physically achieved. Aside from this, at the decision process and based on the observation component $\beta$ a different utterance-expressions take place for understanding the context resulted from the observation. The resultant information from the interactive utterance expressions are submitted to the long-term memory as schema up-date. A proper combination of these components performs fair specialized behavior. The next subsection will describe and explain such interactive behaviors.

## Interactive Expression

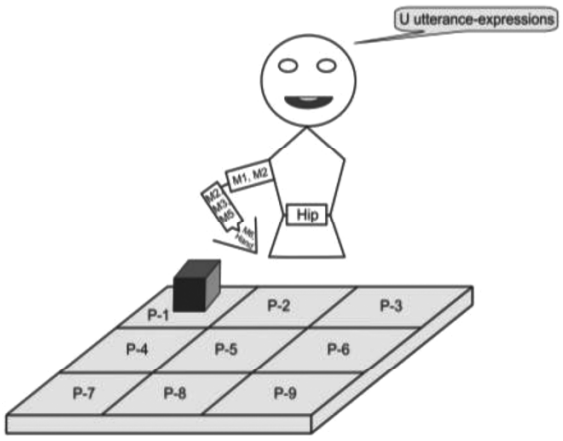
The expressions performed by the robot are mainly low level-behaviors which include processing the streamed signals from the sensors and performing low level interface to the robot's actuators through non-knowledge-based actions. In our approach we propose a higher level behavior that implies the human nature. For example, the nature behavior of the human is to ask why or what about a certain action that we do not understand. This is one example of high level behaviors followed by the human in order to improve our knowledge and understanding about a certain subject. Aside from the low-level robotic behavior which includes signal processing and low-level interface; in our approach the humanoid robot follows this high level human behavior in order to improve its knowledge about a certain situation. In addition, this knowledge improvement occurred by these high level behavior provides another type of high level behaviors which are knowledge-based behavior. The robot behavior at the $k^{th}$ situation is $B_k$ (see Equation (1)) might be described with both motor-expression $M_K$ and utterance-expressions $U_K$.

$$B_K = \{M_K, U_K\}$$
$$M_K = \{m_1, m_2, m_3, \ldots, m_i\}$$
$$U_K = \{u_1, u_2, u_3, \ldots, u_j\} \quad \ldots(1)$$

Motor-expression is a component that provides the mapping from high-level knowledge-based task planning to low-level motor commands that are physically realized while executing these plans. It consists of a collection of trajectories of the motors angles for the robot motors $m_i$ at any step of the task in low-level parameters that provide a task

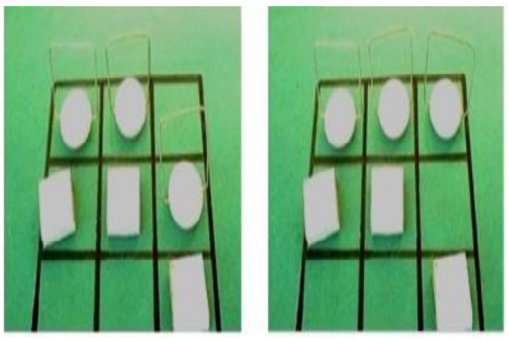
execution to be done. For example, if the robot decides to move the object shown in Figure 3 to a different position such as position $p$ – 3; the motor expression component maps between the trajectories patterns collection for each motor starting from shoulder motor to the hand motor in addition to the hip motors in order to execute such a task. However, if the robot decides to move the same object to a different position such as position $p$ – 9, the motor expression component maps a different trajectory pattern in order to execute the new task. A source position of an object at a situation may become a target position at another situation. Therefore, and in order to avoid code duplication or computing reverse kinematics for performing such a conflict; a plug in is loaded to specify the parameters of the movement in order to adopt the robot gripper to solve such situations. The utterance-expression is a collection of individual spoken words ui at any step of the task.

**Figure 3: Motor and Utterance Expression**




Also mapping between the words depends on the situation itself and also depends on the teacher's answers. In our proposed architecture, since the other components such as decision making or long-term component have the knowledge of task state contextual information, the developer responsible for utterance-expressions does not need to worry about this contextual information. A proper combination of the individual motor-expression and the spoken words ui produces high-level interactive behavior while interacting with the robot. Mapping and selecting the motor-expression $M$ or utterance-expressions $U$ is done by the Decision-making based on the information stored at the long-term memory; also the conclusion depends on the environment events computed by the perception component. An instantiation example of our proposed architecture applied on teaching the humanoid robot *X-O* game strategy is shown in Figure 4. Figure 4a shows a wining chance for the humanoid robot by assembling a line of its round game pieces.

**Figure 4: Teaching Example**



(a) (b)

***Hoap-3:*** Why did you move my piece not your piece, Is this a wining chance?
***Human teacher***: yes
***Hoap-3***: Is this my wining ?
***Human teacher*** : Yes
***Hoap-3*** :That is very good, Please reset the game set to start over again and press enter after you play

At this state the human teacher performs an interruptive action $\delta$ by moving the robot's game piece in order to assemble the line as Figure 4b shows. At the present state of Figure 4b a set of information $X_K$ about the present state is while $X_{K-1}$ holds information about the past state of the task shown in Figure 4a and is stored at the short-term memory $\lambda$ (Figure 2). An observation component $\beta$ takes place between the past state information $X_{K-1}$ and the present state information $X_K$ at the humanoid turn informing the decision making process $\Psi$ about the interruptive action $\delta$ made by the teacher and its classification. The decision making maps between the internal expressions components based on the obtained information from the observation component $\beta$ orchestrating the interactive structured interview as shown in Figure 4. Through the structured interview the humanoid robot is enabled to obtain the goal $\breve{g}$ of the interruptive action $\delta$ of its teacher. In the following sections the formulation of the learning and teaching process is discussed.

## FORMULATION OF OUR ARCHITECTURE

### Decision Making

In this section, we will formalize the notion of a decision making process. First, the decision making process is represented as the following 6 terms tuple.

Decision making $\Psi$:

$$\Psi = (S, \Phi, \beta, \tau, \sigma, \rho) \quad ...(2)$$

with following components.

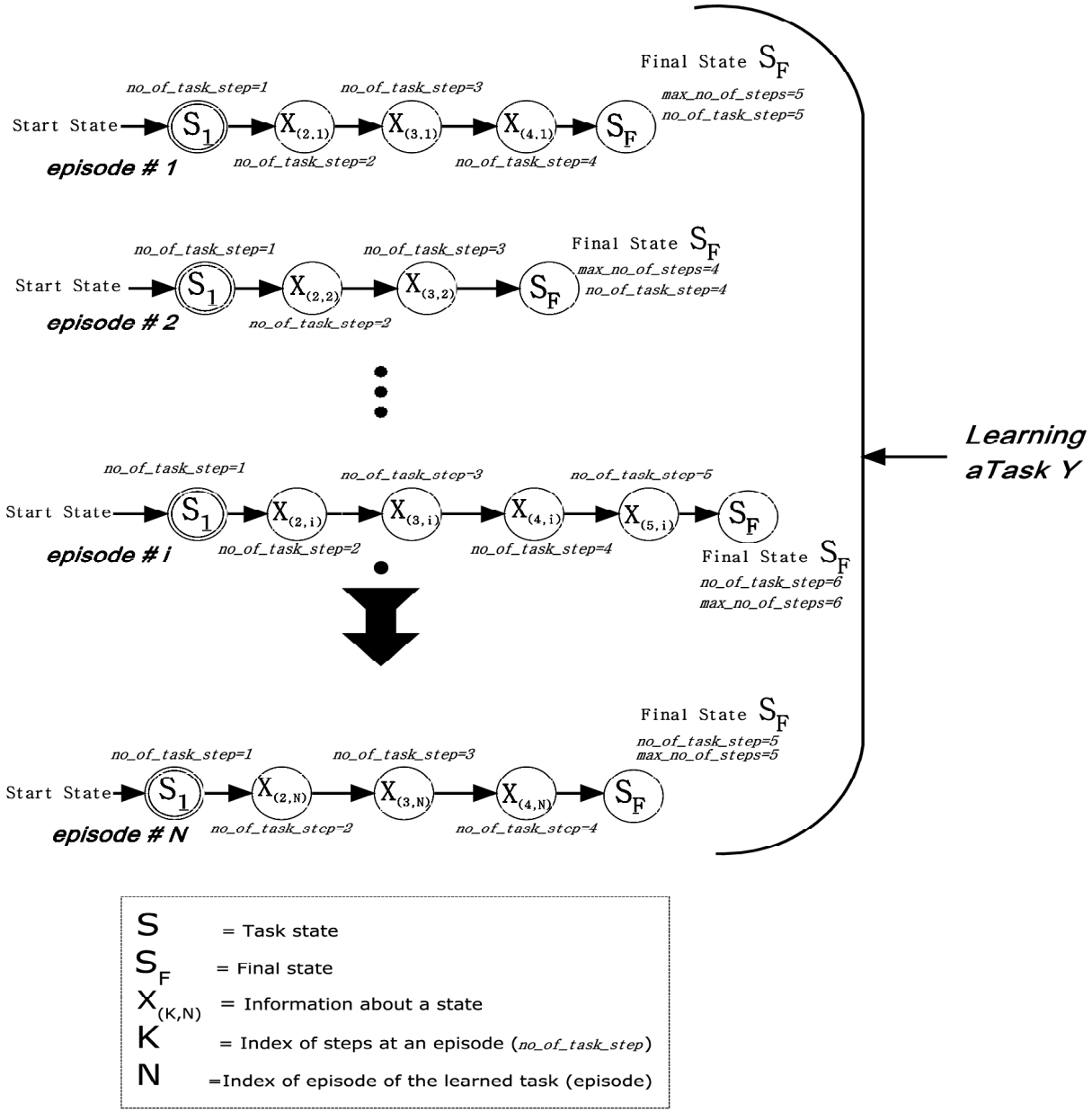
- $S$ is a finite set of states of the learned task,
- $\Phi$ is the stored archive data,
- $\beta$ is an observation component,
- $\tau$ is a process tacking component,
- $\sigma$ is hypothetical scenario component, and
- $\rho$ is the knowledge learned by the humanoid robot.

During learning a task $Y$, every state $S$ is indexed as no_of_task_steps at every episode as shown in Figure 5. Every episode is a new teaching trail between the human teacher and the humanoid robot. Also episodes may end by executing one of the learned actions at the final state $S_F$. At every state of any episode there is a set of information $X$ for representing the situation; these information $X$ about every state in every episode are stored as archive data $\Phi$ as shown in Equation (3).

$\Phi$ = {episode # 1, episode # 2, episode # 3, …, episode # $N$} ...(3)

The archived data finally forms a two dimensional matrix whose column dimension is the no_of_task_steps performed during learning a task $Y$, and its rows dimension is the episode index $N$. New episode starts from the home position $S_1$ of the learned task. Every episode ends at $S_F$ after teaching the humanoid robot a new idea about the learned task $Y$ or if the humanoid robot performs one of the actions taught by its teacher. Notice that, the matrix formed by $\Phi$ is not a square matrix. In other words, the two dimensions of the stored archive data may not be equal. In order to orchestrate a suitable interactive behaviour in response to the information about the present state $X_K$, the decision-making process $\Psi$ subscribes to the information resulted from the observation component $\beta$ and to the archive $\Phi$ in order to perform a process tracking as will be explained in the next section.

**Figure 5: Learning a Task *Y***




## Process Tracking

The decision making performs process tracking procedure $\tau$ by intersecting the present state information $X_K$ (where ($X_K \in S$) with the archive data as shown in the following Equation (4);

$$\tau = X_K \bigcap \Phi \quad ...(4)$$

For example, in order to teach a robot a task strategy such as Connect Four game, at first the human plays then he prompts the humanoid robot the play. The robot processes the present state $X_K$ and stores it at the archive data $\Phi$ with its indexes coordinates which are the (no_of_task_steps index ($K$),

episode index (*N*)). In order to respond present state $X_K$, the decision making $\Psi$ performs a processing to starting from low-level processing (Mahnmoud *et al.*, 2010) at which the robot is able to identify the game piece's shape and coordinates according the 2-D camera frame resulting a 3*3 matrix $X_{matrix\ of\ the\ present\ state}$ which contains nine values, that is three are of value 15 ("hoap-3") and three of value 2 (user) and three zero (empty) as follows.

| | | |
|---|---|---|
| 15 | 15 | 15 |
| 0 | 0 | 0 |
| 2 | 2 | 2 |

Which results the game matrix $X_{matrix\ of\ the\ present\ state}$ (human_parts(2),robot_parts(15)) at the state $S_1$ for example, and is indexed as follows;

| | | |
|---|---|---|
| 1 | 4 | 7 |
| 2 | 5 | 8 |
| 3 | 6 | 9 |

Also the decision making $\Psi$ infers information $X_K$ such as the game pieces types and their numbers also which piece has been moved and the coordinates of its source and target positions. However for faster processing our architecture does not perform process tracking for all the information $X_K$ of the state, it only performs process tracking on the game pieces coordinates matrix Xmatrix of the present state (where $X_{matrix\ of\ the\ present\ state} \in X_K$). The process tracking component is activation range starts from the second episode (episode # 2) after the robot has learned an idea at the first episode # 1 and stops at the one just before the present episode < # *i* which the process tracking is taking place at it. However if the observation component $\beta$ has provided the decision making $\Psi$ with an information about any interruptive actions $\delta$ performed by the teacher; the process tracking also will not be activated. The process tracking subscribes to the archive data $\Phi$ in order to read the matrix $X_{matrix\ from\ the\ archive\ data}$ of every state at every episode (row of the archive matrix $\Phi$) starting from the first state $S_1$ until the end state $S_F$ of every episode (row) resulting a set of matrices information about the all the states in the archive data $X_{matrix\ from\ the\ archive\ data}$ (*H*)(where (*H*) is the number of the read matrix > 0). By logically 'And' the read matrices about the states in the archive data $X_{matrix\ from\ the\ archive\ data\ (H)}$ one by one and the present state matrix $X_{matrix\ of\ the\ present\ state}$ we are able to make intersection. The results from the intersection are $X_{intersected\text{-}\_matrix(h)}$ (where (*h*) is the number of intersected matrices $\geq$ 0) which are the matrices at the archive data that have information resemble to the present state information. The coordinates of the intersected states are organized and stored in the form of two vectors; one vector holds the indexes of the episode that contains the intersected matrix $X_{intersected\ matrix}$ and the other vector holds the index of the no_of_task_steps of the same intersected matrix. The decision making $\Psi$ uses these vectors for performing a hypothetical scenario $\sigma$ with the aid of the information available at long-term memory $\rho$ which are stored during the interaction with the human teacher.

## Hypothetical Scenario $\sigma$

This resulted information from the process tracking $\tau$ provides $\Psi$ the necessary information in order to perform hypothetical

scenario $\sigma$, which is the resultant data from the union of process tracking resulted information $\tau$ and knowledge data $\rho$ as shown in Equation (5). This enables the robot to predict and decide the new step of the task which as follows;

$$\sigma = \tau \bigcup \rho \quad ...(5)$$

The decision making process performs a union between the process tracking $\tau$ and knowledge data $\rho$ based on the information obtained from the tracking process and submitted to it as the two vectors that hold the indexes of the intersected matrix $X_{intersected\ matrix}$. The decision making process reads these information and subscribes to the information at the long-term memory using the indexes from these two vectors. Thus, at some situation's indexes; the knowledge data $\rho$ may submit (null) to the decision-making process $\Psi$. The reason for this is that at these situation's indexes read from the process tracking two vectors; the teacher did not teach the robot any type of goals. The hypothetical scenario $\sigma$ starts to read the information from the knowledge data as available as many as the number ($h$) of the intersected matrixes $X_{intersected\ matrix\ (h)}$. The imported data are all the available data about the states $X_{intersected\ matrix\ (h)}$ that intersected with the present state $X_{matrix\ of\ the\ present\ state}$; such as its indexes (no_of_task_steps index ($K$), episode index ($N$)). Also the max_no_of_steps (Figure 5) which is the episode length $L$ also is imported from the knowledge data. In addition to the flags of the interview which took place between the teacher when teaching the humanoid robot. The concluded results from the union between the process tracking $\tau$ and knowledge data $\rho$ is a new refined vector of structure type that contains schema blocks of information about only the situations $X_{intersected\ matrix\ (J)}$ where ($J$) is the number of states which intersected with the matrix of the present state $X_{matrix\ of\ the\ present\ state}$ and also at which the teacher taught the humanoid robot how to achieve a goal; $J \geq h$ This structured vector of data blocks is submitted to the Adaptive_Greedy_Algorthim in order to perform a rational decision.

## Knowledge Data $\rho$

The knowledge information at the long-term memory is a form of structured vector of schema segments. Each segment includes all the available information about the taught state such as the flags $F^{I}_{(\text{episode \# } i)}$ resulted from the structured interview took place between the teacher and the robot which provide the decision making $\Psi$ with the contextual purpose of the teacher interruptive actions $\Delta$. The segment also contains the spatial coordinates $M_{-(G-P)}$ of the moved pieces. However only the last two or three states (depending on the learning situation) of task matrixes ($X_{L-1}$, $X_L$) of the learned episode are included within the segment. The final value of the variable no_of_task_step which is the episode length $L$ is changed into a new variable named max_no_of_steps (Figure 5) and included at the schema segment block of the learned episode. The knowledge segments are updated at the end of every episode during learning a task. A simplified representation of the knowledge data is shown in Equation (6) as follows;

$$\rho = \{[\delta_1, (X_{1(L-1)}, X_{1(L)}), F^{I}_{1}, (M_{(G-P)}), L_1]_{\text{episode \# 1}},$$
$$[\delta_2, (X_{2(L-1)}, X_{2(L)}), F^{I}_{2}, (M_{(G-P)}), L_2]_{\text{episode \# 2}},$$

$$\vdots$$

$$[\delta_N, (X_{N(L-1)}, X_{N(L)}), F^l_N, (M_{(G-P)}), L_N]_{\text{episode \# }N}\} \quad ...(6)$$

The knowledge data is formed and increased by the interaction between the teacher and the humanoid robot.

## Human Interruption Action its Observation $\beta$

During a task $Y$ the teacher performs an interruptive teaching actions $\Delta$ at the $K^{th}$ state $X_{(K, i)}$ of episode # $i$ as shown in Figure 6. In our learned task we have main teaching purposes; such as teaching the robot a winning or defence movements ($\delta_{is\_winning}$ or $\delta_{is\_defence}$) for the human teacher or for the robot ($\delta_{my\_winning}$, $\delta_{my\_defence}$) as explained in the next Equation (7).

$$\Delta = \{\delta_{is\_winning}, \delta_{is\_defence}, \delta_{my\_winning}, \delta_{my\_defence}\} \quad ...(7)$$

Our interactive learning architecture first uses active learning in order to understand the situations. Therefore a component for identifying different learning situations must be existed. In many proposed approaches (Brian and Gonzalez, 2008), templates were provided in advance in order to assist the system to recognize the situations contextual of the teaching actions. Also in another proposed approach (Mahnmoud *et al.*, 2008), a knowledge data are provided in advance. However, our architecture extracts the individual low-level behaviour context which leads the robot to the high-level behaviour learning starting from observing the interruptive actions $\Delta$ made by the teacher. For example, in the *X-O* game learning task, when the teacher finds a teaching chance by moving one of the humanoid robot game pieces or not moving any of the pieces at all and then prompts the humanoid robot to play, the humanoid robot processes the present state $X_K$ (Figure 6) and stores it at the archive data $\Phi$, and through the observation component $\beta$ which is activated by the decision making process $\Psi$ and takes place at an episode $i$ between the present state information $X_K$ at the working memory with index ($K$, $i$) and $X_{K-1}$ which denotes to past state with indexes ($K$–1, $i$) (Figure 7) and is stored at the sensory memory $\lambda$ (Figure 2). Starting from low-level processing (Mahnmoud *et al.*, 2010) at which the robot is able to identify the game piece's shape and coordinates according the 2-D

**Figure 6: Interruptive Teaching Actions Performed by the Teacher**

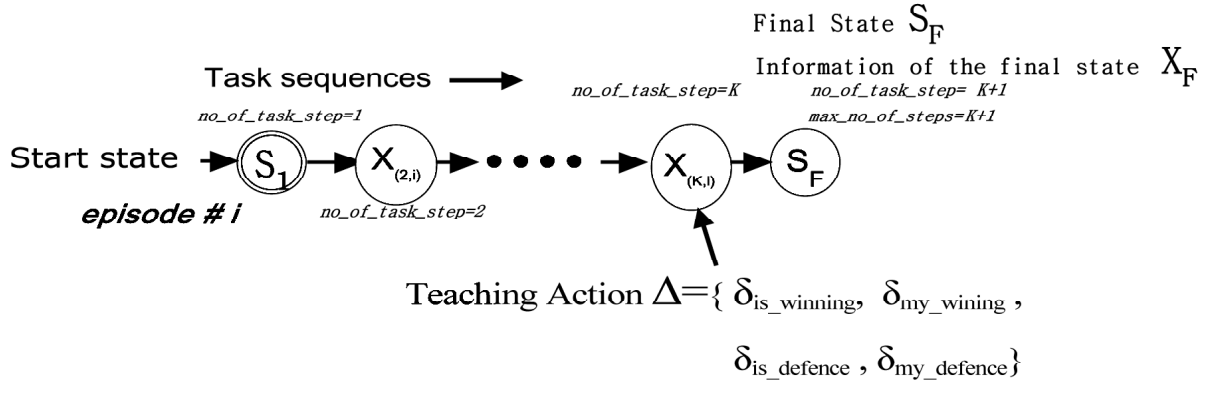
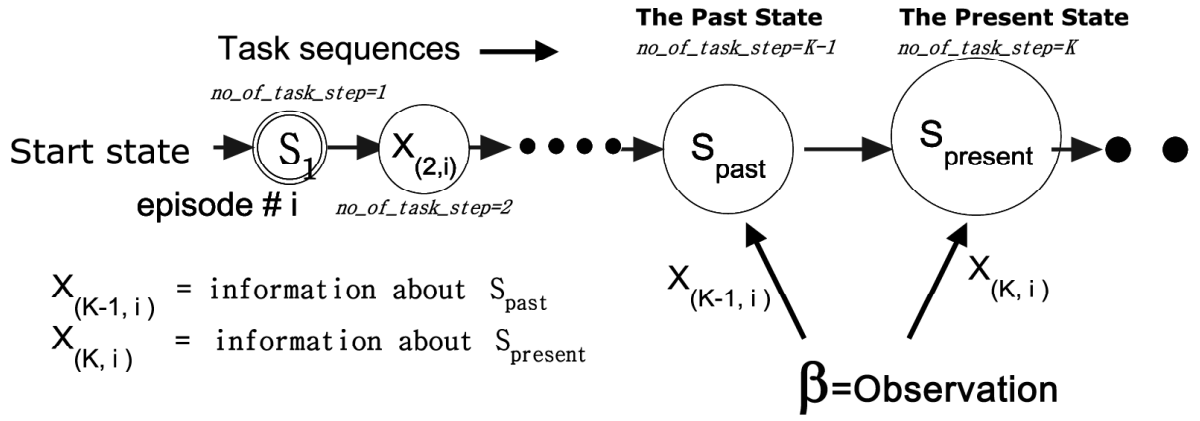


**Figure 7: Observation of the Teaching Actions**




camera frame resulting a 3*3 matrix contains nine values, that is three are of value 15 ("hoap-3") and three of value 2 (user) and three zero (empty) as follows.

| | | |
|---|---|---|
| 15 | 15 | 15 |
| 0 | 0 | 0 |
| 2 | 2 | 2 |

Which results the game matrix X(human_parts(2),robot_parts(15)) at the state $S_1$ for example, and is indexed as follows;

| | | |
|---|---|---|
| 1 | 4 | 7 |
| 2 | 5 | 8 |
| 3 | 6 | 9 |

Among these information is the game matrix which includes the number and the type of the game pieces recognized as X(human_parts, robot_parts). This game matrix is used by the process tracking component $\tau$ (see Equation (4)) in addition to being used by the observation component as the following processes. For faster processing we convert the game matrix X(human_parts, robot_parts) into logical matrix. The human game pieces are changed into zeros, which results in a new matrix $X_{K-(\text{"Hoap-3"-Part})}$ that includes only the robot game pieces as follows;

$X_{K-(\text{"Hoap-3"-Part})}$ (human = 0, robot)

Then the robot's game pieces are changed into ones in which $X$ becomes a logical matrix $X_{K-(\text{"Hoap-3"-Part})}$ and includes only the spatial coordinates of robot's game pieces at $K$ step as follows;

$X_{K-(\text{"Hoap-3"-Part})}$ (human = 0, robot = 1)

On the other hand the same processes are performed to the same original matrix X(human_parts, robot_parts) but for the teacher's game pieces, and produces a new logical matrix $X_{K-(\text{Teacher-Part})}$ which includes the spatial coordinates of the teacher's game pieces at the same $K$, as follows;

$X_{K-(\text{Teacher-Part})}$ (human = 1, robot = 0)

Also the same processes are being performed to the data $X_{K-1}$ which denotes to the information about the past state ($K$–1, $i$) which is stored at the sensory $\lambda$ resulting in new two matrixes, the first one is logical matrix $X_{(K-1)-(\text{"Hoap-3"-Part})}$, includes only the spatial coordinates of robot's game pieces at $K$–1 step, and another logical matrix $X_{(K-1)-(\text{Teacher-Part})}$

and includes only the spatial coordinates of teacher's game pieces at the same step $K$–1 as follows;

$X_{(K-1)-(\text{"Hoap-3"-Part})}$ (human = 0, robot = 1)

and

$X_{(K-1)-(\text{Teacher-Part})}$ (human = 1, robot = 0)

In order to obtain the context of the low-level behaviour performed to the task situation is to process both of the resulted data in $X_K$ and $X_{(K-1)}$ in a logically using Ex-or logic principle as in the syntax followed in the two Equations (8) and (9);

$Đ1_{\text{Teacher-Part}} = (X_{K-(\text{Teacher-Part})}$ (human = 1, robot = 0)

$EX\text{-}OR\, X_{(K-1)-(\text{Teacher-Part})}$ (human = 1, robot = 0) ...(8)

and

$Đ1_{\text{"Hoap-3"-Part}} = (X_{K-(\text{"Hoap-3"-Part})}$ (human = 0, robot = 1)

$EX\text{-}OR\, X_{(K-1)-(\text{"Hoap-3"-Part})}$ (human = 0, robot = 1) ...(9)

The resultant data of this procedure is called an observation data $\beta$ as shown in Equation (10);

$\beta = < Đ1_{\text{Teacher-Part}}, Đ1_{\text{"Hoap-3"-Part}} >$ ...(10)

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= "Teacher" piece has been moved>, if

$\beta = <1, 0>$

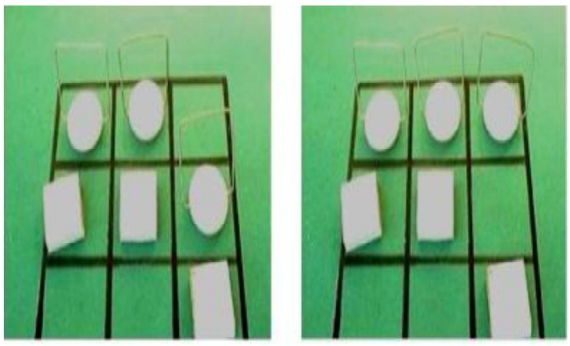
In addition to this, the observation component at our architecture is able to identify the spatial coordinates of the game piece which has been moved by the teacher during the example of Connect four game. These bundles of data are submitted to the Decision-making process as will be explained at the next section, in which the appropriate action is to be selected such as activating the text to speech component for performing a proper structured interview with teacher about the observed situation.

## TESTING AND EVALUATING OUR ARCHITECTURE

We have performed an experiment composed of 100 episodes on teaching a humanoid robot an *X-O* game strategy in which various interactive situations have been taken place, and among these situations. Situation (A) at which a winning chance is available for the robot as in Figure 8a. However as there is no any data provided in advance, the robot will not be able to recognize it. Human teacher performs an interrupting step by moving the robot's game piece instead of his game pieces to set the winning row as in Figure 8b, and 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(human_parts, robot_parts), which is stored as a archived data $\Phi$. On the other hand a single piece of data $X_{K-1}$ is stored at sensory

memory $\lambda$ which denotes to per-performed ($K$–1, $i$) information step, which in our present situation is the matrix in Figure 8a. After applying the observation component in Equations (8) and (9) which leads to higher level observation $\beta$ at Equation (10), the following $\beta$ is obtained;

**Figure 8: Teaching the Humanoid a Wining Chance**



(a) (b)

***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.

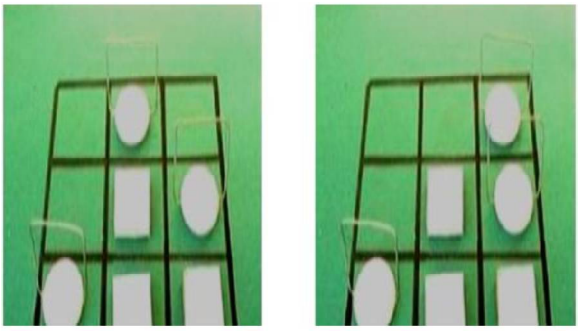
$$\beta = <0, 1>$$

This information is submitted to the Decision-making procedure $\Psi$, which orchestrates a number of individual motor-expressions M such as moving its upper-torso in addition to its arm in order to face its teacher. In addition Decision Making $\Psi$ orchestrates an utterance-expressions $U$ as a structured interview with its teacher in order to obtain the high-level contextual of the teacher interruptive actions. The structured interview utterance-expression provides the necessary information in the form of vector of logical flags as the following equation shows which purify the purpose of the human-teacher action $\Delta$.

$$F^l_i = [\delta_{is\_winning} = 1, \delta_{is\_defence} = 0, \delta_{my\_winnin} = 1, \delta_{my\_defence} = 0]$$

As the structured interview shows, the robot asks the human teacher to reset the game set in order to restart a new episode. On the other hand the Decision-making $\Psi$ submits the information about this situation to the knowledge $\rho$ as a schema block including the last two matrixes of this situation. Also the structured interview flags $F^l_i$ are submitted to within the schema block as a knowledge data

**Figure 9: Teaching the Humanoid Defence Action**



(a) (b)

***Hoap-3***: Why did you move my piece not your piece, Is this a wining chance?.
***Human partner***: No
***Hoap-3***: You mean this is defense step, right?, Is this my defense?
***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.

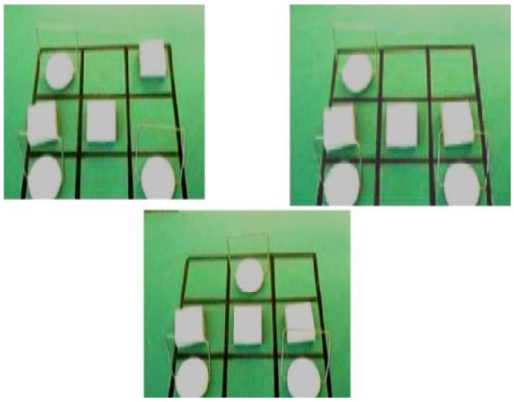
$\rho$. Noting that the index no_of_task_step is submitted as max_no_of_steps within the schema block.

Situation (B) is a chance to teach the robot how makes a defence step is available as shown in Figure 9. For this situition the same procces starting from X(human_parts, robot_parts), until updating the knowledge $\rho$ by subnmitting a block of infroamtion about the sitiauiton takes place. However the structured interview flags $F^I_i$ are differing as follows;

$F^I_i = [\delta_{is\_winning} = 0, \delta_{is\_defence} = 1, \delta_{my\_winnin} = 0, \delta_{my\_defence} = 1]$

Situation (C), at which the human teacher aims to teach the robot the form of his winning as Figure 10 shows in Figure 10a. Human plays and gets the available winning chance as in Figure 10b, 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 be able to recognize it and starts to play randomly and may be as in Figure 10c, 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 $\beta$ and results as follows;

**Figure 10: Teaching the Humanoid How the Human Wining is to Done**



***Hoap-3***: Why did not you move any piece? Is this a wining chance?.
***Human partner***: YES
***Hoap-3***: Is this my wining?
***Human partner***: NO
***Hoap-3***: That is very bad, please reset the game set to start over again and then press enter after you play.

$\beta = <0, 0>$

This means there is no any of the game pieces have been moved. This data is submitted to decision making main frame $\Psi$ which orchestrates a new proper structure interview based on the real-time interaction. Also the same procedure is followed by the robot. The resultant data from the structured interview $F^I_i$ is differing from the previous two situations as follows;

$F^I_i = [\delta_{is\_winning} = 1, \delta_{is\_defence} = 0, \delta_{my\_winning} = 0, \delta_{my\_defence} = 0]$

Also another different in this situation is that the data $X$ that submitted to knowledge updating $\rho$ Includes the matrix Figure 10a. 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 $\Psi$ performs a different procedure from the previously explained situation.

## ADAPTIVE GREEDY ALGORITHM

The first procedure is performing process tracking $\tau$ by intersects the present matrix $X_K$ with the archive data $\Phi$ as explained in Equation (3) (Figure 2). If $\tau$ = <null>, the robot plays randomly (Figure 2). However, if $\tau \neq$ <null> the robot unites the resultant data from $\tau$ with the knowledge data $\rho$.

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

$$F^{I}_{i} = [\delta_{is\_winning} = 1, \delta_{is\_defence} = 0, \delta_{my\_winning} = 1, \delta_{my\_defence} = 0]$$

If the hypothetical scenario $\sigma > 1$, then the robot's final decision $\Psi$ is by choosing an action resembles the episode which has the minimum difference between max_no_of_steps_played and no_of_steps_played of the $X_K$ at which its main priority is achieved.

$$Difference_{min} = max\_no\_of\_steps - no\_of\_task\_steps$$

The second priority is given to:

$$F^{I}_{i} = [\delta_{is\_winning} = 0, \delta_{is\_defence} = 1, \delta_{my\_winning} = 0, \delta_{my\_defence} = 1]$$

Also if the hypothetical scenario $\sigma > 1$ "Hoap-3" final decision $\Psi$ is by choosing an action resembles the episode which has the minimum difference between max_no_of_steps_played and no_of_steps_played of the $X_K$.

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

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

Among the individual expressions 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.

## RESULTS

In order to show the efficiency of our proposed architecture, we performed an experimental test composed of 100 episodes and its sample space is as shown in Table 1. New episode occurs if the robot learns new idea about the winning or defense 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

**Table 1: Experiment Results**

| Order of the 10th 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 |

increasing the number of interactive episodes. This is a clue for improving robot knowledge of the game strategy.

## DISCUSSION

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

### 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.

### 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 is used to update "Hoap-3" knowledge. The design of the interactive architecture can significantly facilitate the implementation of human-robot interactive0scenarios. We will now reflect some design issues on our robot architecture from two perspectives: component design and communication of information between components.

### Flexibility

In order to generate a software architecture that allows multiple applications interactions, isolating the application-specific components is strongly recommended. In our architecture, vision tasks were isolated to a single module. Knowledge of the rules of the game is configured on-line through the structured interview. Also interactive communications components are isolated to different modules. Through this, "Hoap-3" was enabled to interact with other applications using the same components. In the course of the *X-O* game, the rules of the game are configured on-line through structured interviews. This allows us to reconfigure the learning architecture for another game application without code modifications accompanied with the same structured interview sets also. However, only the vision tasks need to be adopted to be fit with the new applications, without affecting the other components of the architecture.

### Synergies

A fast component that publishes information consumed by many other components tends

to improve overall system performance. However on account of multiple lags resulted from processing delays and hardware facilities, both component level robustness and exploiting multiple sources of information are needed to perform efficient input information about for learning. For example, vision components may suffer0from false positive detection events. However, by checking and confirming the input data through confirmation system, it0is possible to identify and avoid such false positive events. A false game pieces positions event0in the *X-O* game checked applying this confirmation, observation component results are checked too. By applying this confirmation system, all errors are recovered.

## REFERENCES


1. Baddeley A D (1996), "Human Memory: Theory and Practice", Psychology Press, Hove.
2. Barry Brian Werger (2000), "Ayllu: Distributed Port-Arbitrated Behaviour-Based Control", in Proc., the 5th Intl. Symp. on Distributed Autonomous Robotic Systems, pp. 25-34. Knoxville, TN.
3. Bransford J, Brown A and Cocking R (2001), "How People Learn", *Brain, Mind, Experience, and School*, Expanded Version, p. 33, National Academy Press, Washington, DC.
4. Brian S and Gonzalez J (2008), "Discovery of High-Level Behaviour from Observation of Human Performance in a Strategic Game", in Systems and Humans, *IEEE Transactions*, Vol. 38, No. 3, pp. 855-874.
5. Cimatti A, Pistore M, Roveri M and Traverso P (2003), "Weak, Strong, and Strong Cyclic Planning via Symbolic Modelhecking", *Artif. Intell.*, Vol. 147, Nos. 1/2, pp. 35-84.
6. Fikes R and Nilsson N (1971), "STRIP: A New Approach to the Application of Theorem Proving to Problem Solving", *Artif. Intell.*, Vol. 2, Nos. 3/4, pp. 189-208.
7. 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.
8. Lauria S and Bugmann G (2002), "Mobile Robot Programming Using Natural Language", *Robotics and Autonomous Systems*, Vol. 38, Nos. 3-4, pp. 171-181.
9. Lockerd A and 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.
10. Maes P and Brooks R A (1990), "Learning to Coordinate Behaviours", in *Pros AAAI*, pp. 796-802, Boston, MA.
11. Mahadevan S and Connell J (1991), "Scaling Reinforcement Learning to Robotics by Exploiting the Subsumption Architecture, in Proc. 8th Int. Workshop Machine Learning, pp. 328-337.
12. Mahnmoud R A, Ueno A and Tatsumi S (2008), " A Game Playing Robot that Can Learn from Experience", in HSI'08 on Human System Interactions, pp. 440-445.

13. Mahnmoud R A, Ueno A and Tatsumi S (2011), "A Game Playing Robot that Can Learn a Tactical Knowledge Through Interacting with a Human" (*ICAART, 2011*), January 28-30, pp. 609-616, Rome, Italy.
14. Marois R (2005), "Two-Timing Attention", *Nature Neuroscience*, pp. 1285-1286.
15. Nicolescu M N and Mataric M J (2001), "Learning and Interacting in Human-Robot Domains", in Systems, Man and Cybernetics, Part A: Systems and Humans, *IEEE Transactions*, Vol. 31, No. 5, pp. 419-430.
16. Penberthy J and Weld D (1992), "UCPOP: A Sound, Complete, Partial-Order Planner for ADL", in Proc. 3rd Int. Conf. KR, pp. 103-114.
17. Reeves B and Nass C (1996), "The Media Equation—How People Treat Computers, Television, and New Media Like Real People and Places", Cambridge University Press, Cambridge, UK.
18. Sweller J (2006), "Visualization and Instructional Design", in 3rd Australasian Conference on Interactive Entertainment, Vol. 207, pp. 91-95.
19. 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.