Building Adaptive Emotion-Based Pet Robots

Wei-Po Lee*, Jyun-Wei Kuo, and Pei-Chun Lai

Abstract—**Designing robots for home entertainment has become an important application of intelligent autonomous robot. Yet, robot design takes considerable amount of time and the short life cycle of toy-type robots with fixed prototypes and repetitive behaviors is in fact disadvantageous. Therefore, it is important to develop a framework of robot configuration so that the user can always change the characteristics of his pet robot easily. In this paper, we present a user-oriented interactive framework that mainly considers the issues of robot control, human-robot interaction, and robot emotion. For evaluation, we use the proposed framework to construct emotion-based pet robots. Experimental results show the efficiency of the proposed framework.**

Index Terms—**pet robot, robot emotion, human-robot interaction, behavior coordination, neural network**

I. INTRODUCTION

In recent years, designing robots for home entertainment has become an important application of intelligent autonomous robot. One special application of robot entertainment is the development of pet-type robots and they have been considered the main trend of the next-generation electronic toys [1][2]. This is a practical step of expanding robot market from traditional industrial environments toward homes and offices.

There have been many toy-type pet robots available on the market, such as Tiger Electronics' Furby, SONY's AIBO, Tomy's i-SOBOT and so on. In most cases, the robots have fixed prototypes and features. With these limitations, their life cycle is thus short, as the owner of pet robots may soon feel bored and no longer interested in their robots. In addition, every individual user expects a personalized robot that has unique characteristics and behaviors, and he can train and interact with the robot in a specific way. Therefore, it would be a great progress to have a framework for robot configuration so that the user can always change the characteristics of his robot according to his personal preferences to create a new and unique one.

Regarding the design of pet robots, there are three major issues to be considered. The first issue is to construct an appropriate control architecture by which the robots can perform coherent behaviors. The second issue is to deal with human-robot interactions in which natural ways for interacting with pet robots must be developed. And the third issue is to include an emotion system with which the pet robot can have different emotions and decide how to behave accordingly. To tackle the above problems, in this paper we present an interactive framework by which the user can conveniently design (and re-design) his personal pet robot according to his preferences. In our framework, we adopt the behavior-based architecture to implement control system for a pet robot to ensure the robot functioning properly in real time. A mechanism for creating behavior primitives and behavior arbitrators is developed in which an emotion model is built for behavior coordination. Different interfaces are constructed to support various human-robot interactions, including device-based control, speech-based control and gesture-based control. To evaluate our framework, we use it to construct a control system for the popular LEGO Mindstorms robot. Experimental results show that the proposed framework can efficiently and rapidly configure a control system for a pet robot, and its human partner can interact with the pet in different ways. In addition, experiments are conducted in which a neural network is used for the pet robot to learn how to coordinate different behaviors for a user-specified emotion model. The results and analyses show that an emotion-based personal pet robot can be designed and implemented successfully by the proposed approach.

II. BACKGROUND AND RELATED WORK

As indicated above, there are three critical issues to be considered in the design of pet robots. The first is the control architecture that affects profoundly on cognition of a nature or artificial system. Researchers have shown that it is not necessary to use complex structures or sophisticated internal representations to obtain intelligent life-like behaviors. Recently, the behavior-based control paradigm has been advocated to build robots [3][4]. The control system is now considered a network of task-achieving control modules (i.e., behaviors) in which the sensor information is connected directly to actions. The design concept of incremental development has also been confirmed to fulfill the important principles of cognitive science and natural evolution [5]. These features are especially important in designing a life-like pet robot that lives with its owner in the physical environment.

In addition to taking an engineering perspective to tackle the problem of robot control, the second critical issue in developing a pet robot is how to deal with the problem of human-robot interaction. There exists a range of interface technologies for interacting with robots [6][7]. Some robots can be operated remotely using interfaces similar to that of radio controlled cars, and some can be controlled through the wired and wireless Internet. Recently, some direct and natural interaction methods are proposed and advocated for service robots, such as speech-based control and gesture-based

---

W.-P. Lee is with the Department of Information Management, National Sun Yat-sen University, Kaohsiung, Taiwan.

J.-W. Kuo is with the Department of Information Management, National Sun Yat-sen University, Kaohsiung, Taiwan.

P.-C. Lai is with the Department of Information Management, National Sun Yat-sen University, Kaohsiung, Taiwan.

*corresponding author. Email: wplee@mail.nsysu.edu.tw

control methods. These methods mean to establish direct contact with robots through their own receivers. Pet robots are service robots that live in the same space with people. The direct and natural interactions between pet robots and human are thus especially essential. With such considerations, specific interfaces for human-robot interactions need to be developed to design.

The third issue to be considered is emotion, an important drive for a pet to present certain level of intelligence [8][9]. In fact, Antonio Damasio has suggested that efficient decision-making largely depends on the underlying mechanism of emotions, and his research has shown that even in simple decision-making processes, emotions are vital in reaching appropriate results [10]. Therefore, our work includes an emotion system to take advantages of cognitive processes to solve the behavior selection problem within the behavior-based control system. By the emotional model, the pet robot can explicitly express its internal conditions through the external behaviors, as the real living creature does. The owner can thus understand the need and the status of his pet robot to then make appropriate interactions with it.

Though there have been several toy-type entertainment robots available, mostly designed with a fixed prototype and characteristics (e.g., [11][12][13]). Sony's robot dog AIBO and humanoid robot QRIO are sophisticated toys with remarkable motion ability generated from many flexible joints. But these toy robots are too expensive to be popular. Also the owners are not allowed to reconfigure the original design. Instead of building an accurate but expensive entertainment robot, in this work, we present a system through which an ordinary robot owner can easily reconfigure his robot, including the changes of robot behaviors, the interacting ways, and the emotion expressions. The system is user-oriented, so the robot owner can decide the characteristics of his robot according to his preferences, and has a unique personal robot to be one's companion.

III. BUILDING EMOTION-BASED PET ROBOTS

A. A User-Oriented Framework

Our aim is to develop a user-oriented approach that can assist a user to rapidly design (and re-design) a special and personal robot. Robot design involves the configuration of hardware and software. Expecting an ordinary user to build a robot from a set of electronic components is in fact not practical. Therefore, instead of constructing a personal pet robot from the electronic components, in this work we take the reconfigurable LEGO Mindstorms robot as the hardware platform, and concentrate on how to imbue a robot with a personalized control system.

Fig. 1 illustrates the system framework of the proposed approach. As can be seen, our system mainly includes three modules to deal with the problems in building a control system for a pet robot. The first module is to design an efficient control architecture that is responsible for organizing and operating the internal control flow of the robot. Because behavior-based control architecture has been used to construct many robots acting in the real world successfully, our work adopts this kind of architecture to design control systems for robots. The second module is about human-robot interaction. As a pet robot is designed to accompany and entertain its human partner in everyday life, interactions between the owner and his pet are essential. Here we develop three ways for human-robot interaction and communication, including device-based (keyboard and mouse), voice-based and gesture-based methods. The third module is an emotion system that works as the arbitration mechanism to resolve the behavior selection problem within the behavior-based control architecture. With the emotion system, a pet robot can act autonomously. It can choose whether to follow the owner's instructions, according to its internal emotions and body states. Our system is designed to be user-oriented and has a modular structure. With the assist of the presented system, a user can build his robot according to his preferences. If he is not satisfied with what the robot behaves, he can change any part of the control software for further correction. Details of each module are described in the following subsections.

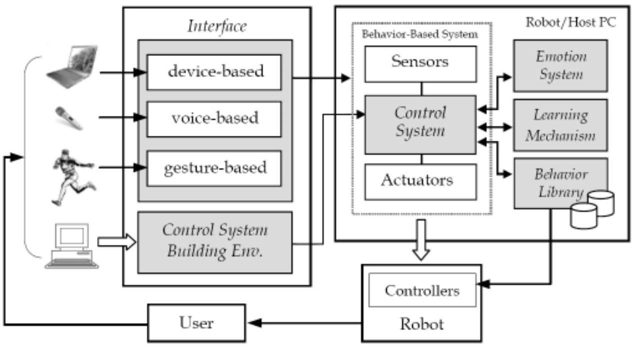


Fig.1: The overall architecture of our system framework

B. Control Architecture

As in the behavior-based control paradigm, the behavior system here takes the structure of parallel task-achieving computational modules to resolve the control problem. In order to achieve more complex tasks in a coherent manner, the behavior modules developed have to be well organized and integrated. Inspired by the ethological models originally proposed to interpret the behavior motivations of animals, (for example, [14][15]) robotists have developed two types of architectures: the flat and the hierarchical ones. The former arranges the overall control system into two levels; and the latter, multiple levels. In the flat type arrangement, all subtasks are independent and have their own goals. Each of them will be achieved by a separate behavior controller, which generates its own output according to the sensory input directly relevant to the subtask itself. Those independent outputs from the separate controllers will be combined appropriately in order to generate the final output for the overall task. As this work means to provide an interactive and easy-to-use framework for the design and implementation of pet robots, the straightforward way (that involves less task decomposition techniques), the flat architecture, is chosen to be the default control structure for a pet.

Using behavior-based control architecture, one needs to deal with the corresponding coordination (or action selection) problem. It is to specify a way to combine the various outputs from the involved controllers. Our work here takes the method of *command switching* that operates in a winner-take-all fashion. That is, only one of the output commands from the involved behavior controllers is chosen to take over the control at any given moment, according to certain sensory stimuli.

As mentioned above, we intend to build low cost toy-type robots as embodied digital pets. Yet, since a toy-type robot only has limited computational power and storage resources, it is generally not possible to perform all the above computation on such a robot. Therefore, to implement the above architecture, an external computing environment is needed to build robots. The experimental section will present how we construct an environment for the LEGO robots.

C. User-Robot Interaction

To communicate with a pet robot, our framework provides two natural ways for interacting with the robot by using oral or body languages. The details are described below.

The most important procedure in using a speech-based approach is speech recognition. In our current implementation, we use the popular speech-control method, command-and-control, to communicate with the robot, and adopt the Microsoft Speech API to implement the speech recognition mechanism in our Windows application.

The first step to add command-and-control voice recognition to an application is to create a voice command object. In this way, when the user speaks command words, the objects corresponding to the words can be activated via function calls. Thus, the user can interact with his robot by speaking to the robot using the pre-recorded command words. Once the command words are recognized, the corresponding controller will be triggered and executed. Though more sophisticated language interface can be developed (for example, [16]), here we simply parse a sentence into individual words, distinguish whether it is an affirmative or negatibe sentence, and then recognize the user commands from the words.

Gesture recognition is another way used in this work to support natural communication between human and robots. Gesture recognition is the process by which the user's gestures are made known to the system via appropriate interpretation. To infer the aspects of gesture needs to sense and collect data of user position, configuration, and movement. This can be done by directly using sensing devices attached to the user (e.g., magnetic field trackers), or by indirectly using cameras and computer vision techniques. Here we take a dynamic gesture recognition approach, and use digital cameras with image processing techniques for gesture recognition.

For simplicity, in our current implementation, we use light spots to represent the hand positions of a pet robot owner, and extract the light track with the cubic spline interpolation method to obtain a behavior curve. Then we take a curve-fitting approach to match the curve produced by the continuous hand movement to the ones recorded previously as user's instructions in the database. Fig. 2 shows a typical example. A successful match means that the gesture has been recognized and its corresponding instruction is then identified. As the traditional approach of dynamic programming for curve fitting has inherent drawback caused by the curve discretization-based distance measurement, in this work we employ the approach reported in [17] that takes the underlying differential equation into account and finds a continuous solution for curve fitting. The fitting result is then used to recognize gestures.

Fig. 3 illustrates our speech/gesture recognition mechanism that mainly includes three parts. The first part is to develop and maintain a behavior library within which different behavior modules are pre-constructed and recorded. The second part is to capture and record an owner's spoken-sentences/images, and to extract the words/curves for further interpretation. The third part is a matching procedure that tries to match the extracted words/curves to the ones recorded in the mapping table in which each command-word/curve represents a behavior controller. If a match is found, the corresponding behavior module is then retrieved and activated from the behavior library. The behavior controller is executed and the control command is used to drive the robot. Alternatively, if the behavior controllers can be pre-uploaded to the on-board computer of the robot, the mapping result will send an identification signal to trigger the corresponding controller on the robot. Here the user is allowed to construct the mapping table to decide how to interact with his robot.

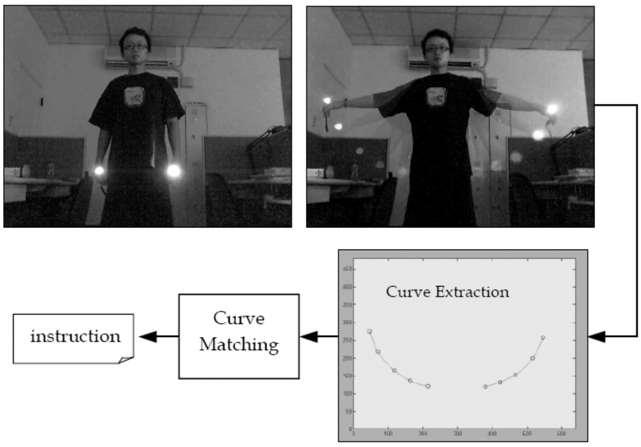


Fig. 2: An example of extracting trajectory curve

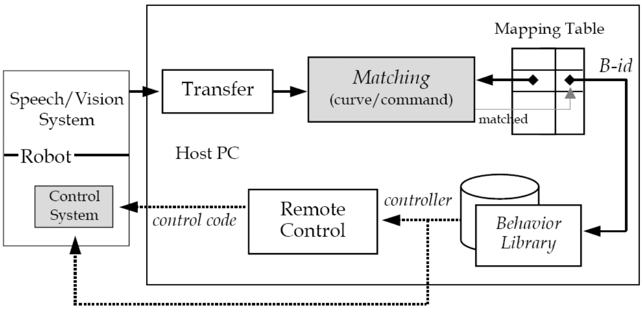


Fig. 3: Speech/gesture-based control

D. Emotion Model

To coordinate different behavior controllers, our framework uses a special mechanism that exploits emotions for selection of behavior. Emotions are often categorized into two kinds: basic emotions and higher cognitive emotions. Basic emotions (such as joy, anger, fear, etc.) are considered as universal and innate; they are encoded in the genome. Higher cognitive emotions (such as love, shame, pride, etc.) are universal too, but they exhibit more cultural variations. Basic emotions tend to be reflex-like responses over which animals have little conscious control. They involve less cognitive processes and are thus much faster in controlling motion than those culturally determined experiences residing in the cognitive system (long term memory). As our goal here is to establish a framework for constructing personal pet robots, rather than to investigate the interactions and relationships between cognitive and emotion systems, our work only models basic emotions to coordinate the behave controllers the user has pre-chosen for his pet. To quantify the emotions, we define each of the basic emotion as: $E_i(t) = E_i(t\text{-}1) + \alpha \times Event_i$, in which $E_i(t)$ represents the quantity of emotion E_i at any time t, α is a user-specified weight, and $Event_i$ is a procedure describing how emotion E_i is affected by a set of events pre-defined by the user. For example, a user can define an event to be the appearance of a stranger. When this happens, "fear" (one kind of emotion of the robot) will increase one unit accordingly. An experienced user can define a more sophisticated procedure to alleviate the effect caused by the same event occurring during a short period of time.

In addition to emotions, another set of internal variables is defined to describe a robot's body states (e.g., hungry). These variables represent the basic requirements the robot has to be satisfied, and they must be regulated and maintained in certain ranges. In our framework, the range of each internal variable can be defined by the user, and the corresponding specification determines the innate characteristics of his robot. Similar to the emotion functions defined above, an internal variable here is described by a mathematical formula with a set of involved events, also specified by the user.

Fig. 4 shows the aspect of the emotion system used. The emotion model determines the innate characteristics of a robot, which are highly related to its abilities of learning and adaptation. With the above model, at any given time, the emotions of the robot can be derived and used to trigger the behavior controller to generate a specific action. After the action is performed, the external environment conditions will thus change and that can further affect the emotion of the robot. The pet robot then makes new decision for behavior selection, based on the modified quantities of emotions. For example, a pet dog in a hungry state may be angry, may keep looking for food, and would eat anything as soon as the pet dog finds it. After that, the dog may not be hungry any more (the body state has been changed). Then it is happy (the emotion has been changed too) and may want to sleep or fool around (now new behavior is selected). The above procedure is repeated and the emotion model continuously works as a decision-making mechanism for behavior selection.

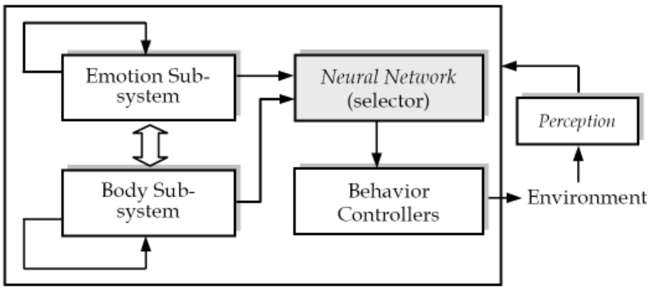


Fig. 4: Emotion-based behavior coordination

As can be observed, in the above operating model, the most important part is the one for selecting appropriate behavior at any time. In this work, we use a feedforward neural network to implement such a mechanism that maps the emotions and body states of a robot into a set of desired behaviors. To allow the user to determine the characteristics of his pet, our work also provides an interface by which the user can define examples to train the neural network to achieve the specified mapping of emotions and behaviors. Once the neural network is obtained, it works as a behavior selector to choose appropriate controllers for the robot.

IV. EXPERIMENTS AND RESULTS

To evaluate our approach, two LEGO robots have been built and a distributed and networked computing environment has been developed for the robots. An example is demonstrated in which two robots cooperate together to achieve a box-pushing task. In the second phase, we show how the emotion-based mechanism can be used to train a neural network as a behavior selector for the robot.

A. Implementation

The robot used in the experiments is LEGO Mindstorms NXT 9797. It has light sensors for light detection, ultrasonic sensors for distance measurement, and micro-switches for touch detection. In addition, to enhance its vision ability, we equip an extra mini-camera on the head of the robot so that it can capture images for further interpretation. Fig. 5 shows the pet robot built for the experiments, in which the robot uses two connected NXTs to collect signals and to execute control programs.

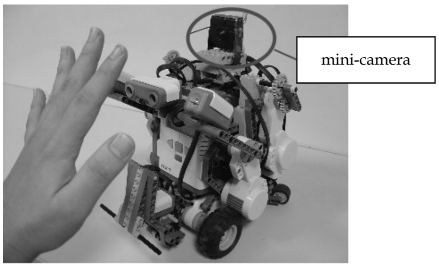


Fig. 5: The LEGO robot with a mini-camera

Because the LEGO NXT only has limited computational power and memory devices, it is thus not possible to perform all the computation for the emotion system and speech/vision processing on the on-board processor of the robot. To implement the architecture presented in section 3.1, a distributed and networked computing environment is constructed for the robot. Figure 7 illustrates the architecture of the computing environment. As can be seen in Figure 7, the individual pre-programmed behavior controllers are installed in the NXT on-board memory, and two PCs are connected through the Internet for other computation. Here, one PC is responsible for the computation of emotion modeling, speech recognition, and communication management, and the other PC is used for image/gesture recognition. The images collected from the mini-camera on the robot are sent to the host PC wirelessly for further processing. After the images are interpreted as described in section 3.3, the result is sent to another PC and used as perceptual information to determine how to command the robot. The network-based computing environment can be extended to deal with more computation when more robots are involved.

The above architecture also enhances the communication ability between different NXT robots. In the original design of LEGO NXT, the robot-robot communication is structured in a one-master-three-slaves manner. A slave NXT can communicate with another NXT by sending a message to the master NXT, which then forwards the message to the target NXT. But in our newly developed computing architecture, the host PC works as an active master that can communicate with NXTs easily through the Bluetooth channel. In this way, the number of slave NXTs can now be extended, depending on the computational power of the host PC.

The emotion system is also executed in one of the host PCs. It interprets the received signals to activate different events to change emotions and body states accordingly, as described in section 3.4. The newly derived values of emotions and states are used to choose a behavior controller, according to the mapping strategy that can be pre-wired manually or learnt automatically. Once the behavior controller is selected, the host PC then sends an identification signal through the Bluetooth channel to evoke the corresponding controller from the set of controllers pre-stored on the NXT robot.

After the above computing environment was established, experiment has been conducted to show how our LEGO robots can achieve a user-specified task. As soon as the owner asked one of the robots to push a red box, it started moving around to look for the box. Once the robot found the box, it tried to push the box. The robot performed a color-recognition behavior controller that took images captured by the camera as input and identified the color of the target by analyzing the RGB values of image pixels. The box was too heavy so that the robot could not push it away alone. The robot then sent a message to another robot through the built-in Bluetooth communication channel to ask for help. The second robot came over, found the box and tried to push it too. As shown in Fig. 8, after the two robots pushed the box together, the red box was moved away successfully.

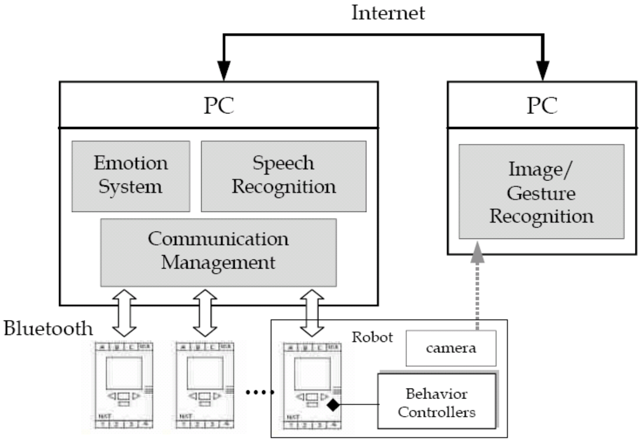


Fig. 7: The computing environment for the LEGO pet robots

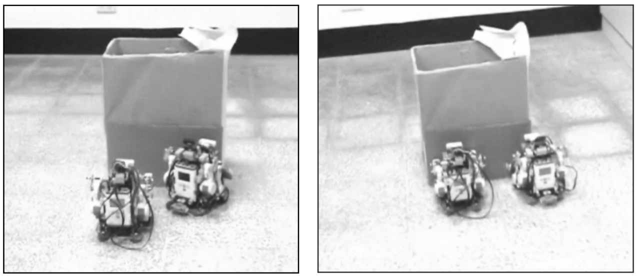


Fig. 8: Two LEGO NXT robots push the red box together

B. *Building Emotion-Based Behavior Selector*

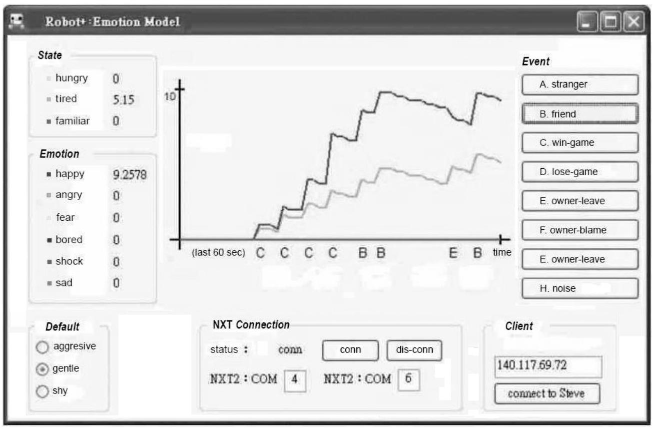
To verify our approach of emotion-based control, this section describes how the emotion model can be built, trained and modified. In the experiments, the simple types of emotions are modeled. They are so-called “basic emotions”, including “happy”, “angry”, “fear”, “bored”, “shock”, and “sad”. They are the most common emotions and may be called by different names in other research work. Also three variables, “hungry”, “tired”, and “familiar” are defined to indicate the internal body states of the robot. As mentioned in section 3.4, the user is allowed to define event procedures and the relevant weight parameters for the above emotions and body states to describe how the quantities of different emotions vary over time for their own robots. For naïve users, our framework also provides three default emotion models (aggressive, gentle, and shy) and events as choices to represent different characteristics of a robot. For example, a robot with an aggressive model will change its emotions rapidly than others. Users can choose a default model from the interface without extra settings. Experienced users can develop more sophisticated models to guide the variations of emotions.

Currently, ten basic behaviors are built, including target seeking, barking, wandering, shaking, sniffing, sleeping, escaping, scratching, wailing, and wagging. As mentioned above, with the emotion model developed, a user can build a behavior selector manually or automatically and use it to map the emotions and body states into appropriate behavior controllers. At each time step, the internal emotions and states of the robot change and the newly obtained values are used as the input of an emotion model to select behavior controller at that moment. Fig. 9 shows the interface that presents the numerical values of the internal and emotion variables over time during a trial. These values are illustrated to provide information about the body state and emotion of the robot, so that the user can inspect the detailed information related to his personal robot accordingly.

As can be seen in Fig. 9, the interface also includes a set of event buttons on the right hand side to simulate the happening of different events. Each button here is associated with an event procedure that describes how emotions and body states are changed by this event. It corresponds to a situated event in reality. That is, when the pre-conditions of an event procedure are satisfied in the real world, the same effect will be given to change the emotions and body states of a robot. With the assistance of event buttons, users can examine the correctness of the effect caused by the event procedure he defined in an efficient way.

In addition, our framework offers a learning mechanism to train a feedforward neural network from examples as a behavior selector. Here, the above variables (emotions and body states) are arranged as the input of the neural network, and the output of the network is used to determine which behavior to perform at a certain time. In the training phase, the user is allowed to give a set of training examples and each specifies which behavior the robot is expected to perform when the set of emotions and states reaches the values he has assigned. The back-propagation algorithm is then used to derive a model for the set of data examples. Fig. 10 presents the interface through which a user can edit the training set. Based on the training set provided by the user, the system then tries to learn a mapping strategy with best approximation.

It should be noted that it is possible the examples provided by the user are inconsistent and consequently a perfect strategy cannot be obtained. In the latter case, the user can use the interface shown in Fig. 10 to correct the training examples to re-build the behavior selector (i.e., the neural network) again. If the model has been derived successfully but the behavior of the robot did not satisfy the owner's expectation, he can still correct the robot behavior for any specific time step by editing the output produced by the mapping strategy learnt previously through the interfaces shown in Fig. 10. Then the modified outputs can be used as new training examples to derive a new strategy of behavior arbitration. In this way, the user can easily and conveniently design (and re-design) the characteristics of his personal robot.



hungry 0
tired 5.15
familiar 0

happy 9.2578
angry 0
fear 0
bored 0
shock 0
sad 0

○ aggresive
● gentle
○ shy

status : conn
NXT2 : COM 4
NXT2 : COM 6

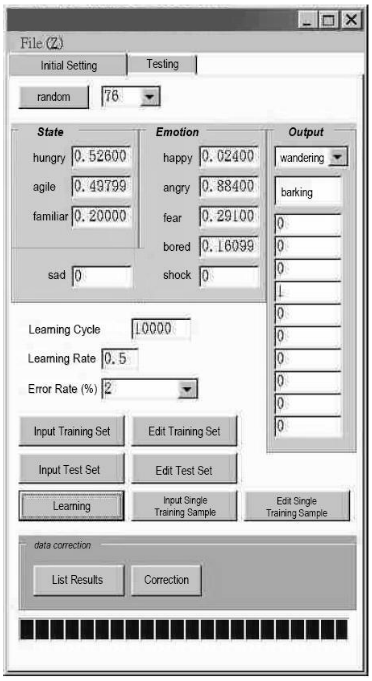
140.117.69.72

Fig. 9: The interface for showing the typical results

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have described the importance of developing toy-type pet robots as an intelligent robot application. We have also proposed to integrate knowledge from different domains to build low-cost pet robots. To realize the development of pet robot, a user-oriented interactive framework has been constructed with which the user can conveniently configure and re-configure his personal pet robot according to his preferences. Our system framework mainly investigates three issues: robot control, human-robot interaction, and robot emotion. Different interfaces have also been constructed to support various human-robot interactions. Most importantly, an emotion-based mechanism has been developed in which different emotions and internal states have been modeled and used to derive a behavior selector. The behavior selector is a neural network and the user is allowed to define training examples to infer a behavior selector for his robot. To evaluate our framework, we have used it to build LEGO NXT robots to achieve a cooperation task successfully. In addition, a neural network has been used for a robot to learn how to exploit the emotion model to work as a behavior coordinator.

Based on the presented framework, we are currently trying different toy-type robots with more sensors and actuators to evaluate our approach extensively. Also we are implementing a new vision module for the robot so that it can recognize human facial expressions and interact with people accordingly. In addition, we plan to define a specific language and construct a message-passing channel through which different types of robots can communicate with each other.



76

hungry 0.52600
agile 0.49799
familiar 0.20000
sad 0

happy 0.02400
angry 0.88400
fear 0.29100
bored 0.16099
shock 0

wandering
barking
0
0
0
1
0
0
0
0
0
0

Learning Cycle 10000
Learning Rate 0.5
Error Rate (%) 2

Fig. 10: The interface for preparing training data

REFERENCES

[1] M. M. Veloso, Entertainment robotics, *Communication of the ACM,* 45(3), 59-63, 2002.
[2] M. Fujita, On activating human communications with pet-type robot AIBO, *Proceedings of the IEEE*, 92(11), 1804-1813, 2004.
[3] R. C. Arkin, *Behavior-Based Robotics*, MA: MIT Press, 1998.
[4] R. A. Brooks, A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation,* 2(1), 1986, 14-23.
[5] R. Pfeifer and C. Scheier, *Understanding Intelligence*, MA: MIT Press, 2000.
[6] S. Thrun, Toward a framework for human-robot interaction, *Human-Computer Interaction,* 19(1-2), 9-24, 2004.
[7] D. Perzarcowski, A. Schultz, W. Adams, E. Marsh, and M. Bugajska, Building a multimodal human-robot interface, *IEEE Intelligent Systems,* 16(1), 16-21, 2001.
[8] J. E. LeDoux, *The Emotional Brain*, NY: Simon & Schuster, 1996.
[9] J.-M. Fellous and M. Arbib (eds.), Who Needs Emotions? The Brain Meets the Robot, Oxford University Press, 2005.
[10] A. R. Damasio, *Descarte's Error: Emotion, Reason, and Human Brain,* NY: Grosset/Putnam Press, 1994.
[11] R. C. Arkin, M. Fujita, T. Takagi, and R. Hasegawa, An ethological and emotional basis for human-robot interaction, *Robotics and Autonomous Systems,* 42(3-4), 191-201, 2003.
[12] T. Ishida, Y. Kuroki, and J. Yamaguchi, Mechanical system of a small biped entertainment robot, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.1129-1134, 2003.
[13] T. Kanda, H. Ishiguro, T. Ono, M. Imai, and K. Mase, Development and evaluation of an interactive robot "Robovie", *Proceedings of IEEE International Conference on Robotics and Automation*, 1848-1855, 2002.
[14] G. P. Baerands, A model of the functional organization of incubation behavior, *Behavior Supplement*, 17, 263-312, 1970.
[15] N. Tinbergen, *The Study of Instincts*, Clarendon Press, 1951.
[16] J. Bos and T. Oka, A spoken language interface with a mobile robot, *Artificial Life and Robotics*, 11, 42-47, 2007.
[17] M. Frenkel and R. Basri, Curve matching using the fast marching method, *Proceedings of International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, LNCS-2683, pp.35-51, 2003.