2nd International Conference on Autonomous Robots and Agents
December 13-15, 2004 Palmerston North, New Zealand

The Third Generation of Robotics: Ubiquitous Robot

Jong-Hwan Kim, Yong-Duk Kim, and Kang-Hee Lee
Robot Intelligence Laboratory, KAIST,
Yuseong-gu, Daejeon 305-701, Republic of Korea
{johkim, ydkim, khlee}@rit.kaist.ac.kr

Abstract

In this paper, ubiquitous robot (Ubibot), a 3rd generation of robotics, is introduced as a robot incorporating three forms of robots: software robot (Sobot), embeded robot (Embot) and mobile robot (Mobot), which can provide us with various services by any device through any network, at any place anytime in a ubiquitous space. Sobot is a virtual robot, which has the ability to move to any place through a network. Embot is embedded within the environment or in the Mobot. Mobot provides integrated mobile services, which are seamless, calm and context-aware. Following its definition, the basic concepts of Ubibot are presented. A Sobot, called Rity, developed at the RIT Lab., KAIST, is introduced to investigate the usability of the proposed concepts. Rity is a 3D synthetic character which exists in the virtual world, has a unique IP address and interacts with human beings through an Embot implemented by a face recognition system using a USB camera. To show the possibility of realization of Ubibot by using the current state of the art technologies, two kinds of successful demonstrations are presented.

Keywords: Ubiquitous robot, Ubiquitous computing, Software robot, Embedded robot, Mobile robot

1 Introduction

In an ubiquitous era we will be living in a world where all objects such as electronic appliances are networked to each other and a robot will provide us with various services by any device through any network, at any place anytime. This robot is defined as a ubiquitous robot, Ubibot, which incorporates three forms of robots: software robot (Sobot), embeded robot (Embot) and mobile robot (Mobot) [4, 1].

The Ubibot is following the paradigm shift of computer technology. The paradigm shift of robotics is motivated by ubiquitous computing and the evolution of computer technology in terms of the relationship between the technology and humans [2, 3]. The basic concepts of ubiquitous computing include the characteristics, such as every device should be networked; user interfaces should operate calmly and seamlessly; computers should be accessible at anytime and at any place; and ubiquitous devices should provide services suitable to the specific situation. Computer technology has been evolving from the the mainframe era, where a large elaborate computer system was shared by many terminals, through the personal computer era, where a human uses a computer as a stand-alone or networked system, in a work or home environment, to the ubiquitous computing era, where a human uses various networked computers simultaneously, which pervade their environment unobtrusively.

Considering the evolution of robot technology, the first generation was dominated by industrial robots followed by the second generation in which personal robots are becoming widespread these days, and as a third generation in the near future, Ubibot will appear. Comparing the paradigm change between the personal robot and ubiquitous robot eras, the former is based on individual robot systems and the latter will be employing multiple robot systems using real time broadband wireless network based on IPv6.

The Ubibot has been developed based on the robot technology and the concept of ubiquitous computing in the Robot Intelligence Technology (RIT) Lab., KAIST since 2,000 [5]. In the future we will live in a ubiquitous world where all objects and devices are networked. In this ubiquitous space, u-space, a Ubibot will provide us with various services anytime, at any place, by any device, through any network. Following the general concepts of ubiquitous computing, Ubibot will be seamless, calm, context-aware, and networked.

This paper presents the definition and basic concepts of Ubibot incorporating three forms of robots; Sobot, Embot, and Mobot. A Sobot, called Rity, developed at the RIT Lab., KAIST, is introduced to investigate the usability of the proposed concepts of Ubibot. Rity is a 3D synthetic character which exists in the virtual world, has a unique IP address and interacts with human beings through an Embot implemented by a face recognition system using a USB camera.

Rity is an autonomous agent which behaves based on its own internal states, and can interact with a person in real-time. It can provide us with an entertainment or a help through various interactions in real life. To realize this, it needs an autonomous function, artificial emotional model, learning skill, sociableness, and its own personality [6, 7]. It can be used as a character on a game or a movie or for the purpose of education [8, 9].

An architecture of Rity can be divided into five modules: perception module, internal state module to implement motivation, homeostasis, and emotion [10, 11, 12], behavior selection module [13, 14], interactive learning module [15], and motor module.

To show the possibility of realization of Ubibot, two kinds of demonstrations are carried out by using the current state of the art technologies.

This paper is organized as follows. Section II presents the definition and basic concepts of Ubibot. Section III describes the overall architecture of the Sobot. Demonstrations of the Sobot, Rity are provided in Section IV. Finally, concluding remarks follow in Section V.

2 Ubiquitous Robot: Ubibot

Ubibot is a general term for all types of robots incorporating software robot (Sobot), embedded robot (Embot), and mobile robot (Mobot) which exist in a u-space. Ubibot exists in the u-space which provides physical and virtual environments.

2.1 U-space and Ubibot

Ubiquitous space (u-space) is an environment in which ubiquitous computing is realized and every device is networked. The world will be composed of millions of u-spaces, each of which will be closely connected through ubiquitous networks. A robot working in a u-space is defined as a Ubibot and provides various services through any network by anyone at anytime and anywhere in a u-space.

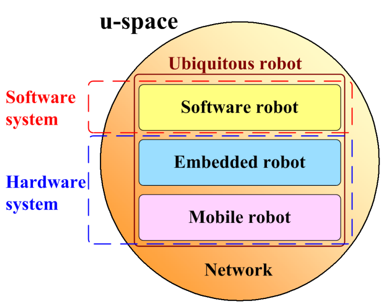


Figure 1: Ubibot in ubiquitous space

Ubibot in a u-space consists of both software and hardware robots. Sobot is a type of a software system whereas Embot and Mobot are hardware systems, Figure 1. Embots are located within the environment, human or otherwise, and are embedded in many devices. Their role is to sense, analyze and convey information to other Ubibots. Mobots are mobile robots. They can move both independently and cooperatively, and provide practical services. Each Ubibot has specific individual intelligence and roles, and communicates information through networks. Sobot is capable of operating as an independent robot but it can also become the master system, which controls other Sobots, Embots and Mobots residing in other platforms as slave units. Their characteristics are summarized in the following. For details, the reader is referred to [1].

2.2 Software Robot: Sobot

Since Sobot is software-based, it can easily move within the network and connect to other systems without any time or geographical limitation. It can be aware of situations and interact with the user seamlessly. Sobot can be introduced into the environment or other Mobots as a core system. It can control or, at an equal level, cooperate with Mobots. It can operate as an individual entity, without any help from other Ubibots. Sobot has three main characteristics, such as self-learning, context-aware intelligence, and calm and seamless interaction.

2.3 Embedded Robot: Embot

EmBot is implanted in the environment or in Mobots. In cooperation with various sensors, Embot can detect the location of the user or a Mobot, authenticate them, integrate assorted sensor information and understand the environmental situation. An Embot may include all the objects which have both network and sensing functions, and be equipped with microprocessors. Embots generally have three major characteristics, such as calm sensing, information processing, and communication.

2.4 Mobile robot: Mobot

Mobot is able to offer both a broad range of services for general users and specific functions within a specific u-space. Operating in u-space, Mobots have mobility as well as the capacity to provide general services in cooperation with Sobots and neighboring Embots. Mobot has the characteristics of manipulability by implementing arms and mobility which can be implemented in various types, such as wheel and biped. Mobot actions provide a broad range of services, such as personal, public, or field services.

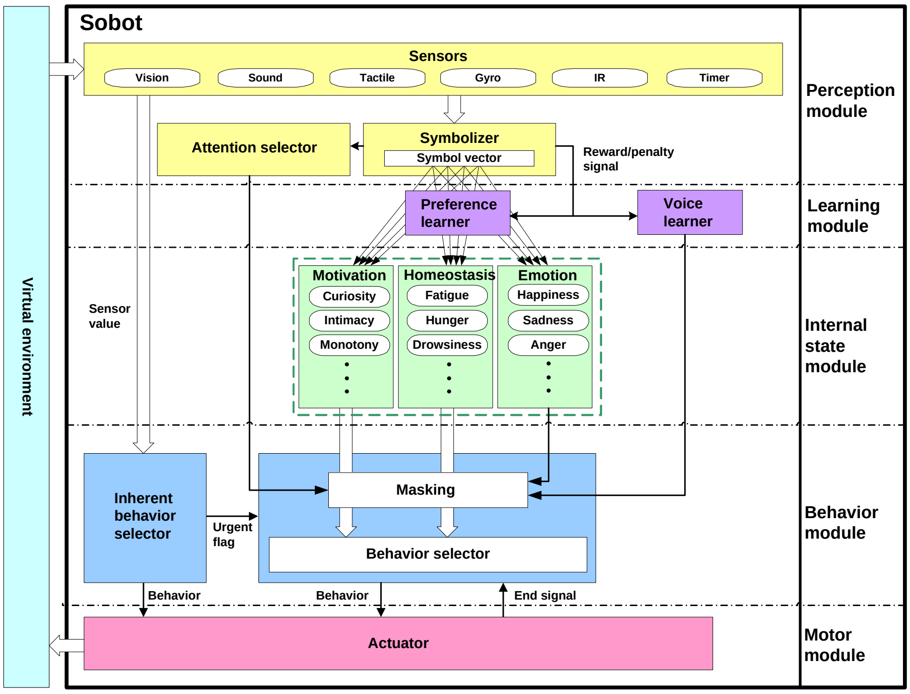


Figure 2: Overall architecture of Rity

3 Implementation of Sobot

Sobot is a software robot which recognizes a situation by itself, behaves based on its own internal state, and can interact with a person in real-time. Sobot should be autonomous; it must be able to select a proper behavior according to its internal state such as motivation, homeostasis and emotion. Also, Sobot should be adaptable; it should adapt itself to its environment. For the purpose of achieving these functions easily and efficiently, Sobot mimics an animal which is an autonomous and adaptable agent in nature. Fig. 2 shows an overall architecture of the proposed Sobot, Rity, where necessary modules are defined as follows: 1) Perception module, which perceives environment through virtual and physical sensors, 2) Internal state module, which includes motivation,homeostasis and emotion, 3) Behavior selection module, which selects a proper behavior, 4) Learning module, which learns from the interaction with a people, and 5) Motor module, which executes a behavior and expresses emotion.

3.1 Perception module

The perception module includes a sensor unit, a releaser having stimulus information provided by a symbol vector and a sensitivity vector, and attention selector. This module can perceive and assess the environment and send the stimulus information to the internal state module. Sobot has several virtual sensors for light, sound, temperature, touch, vision, gyro, and time. Sobot can perceive 47 types of stimulus information from these sensors. Based on these information, Sobot can perform 77 different behaviors.

3.2 Internal state module

The internal state module defines the internal state with the motivation unit, the homeostasis unit and the emotion unit. Motivation (M) is composed of six states: curiosity, intimacy, monotony, avoidance, greed, and the desire to control. Homeostasis (H) includes three states: fatigue, hunger, and drowsiness. Emotion (E) includes five states: happiness, sadness, anger, fear, and neutral. According to the internal state, a proper behavior is selected.

3.3 Behavior selection module

Behavior selection module is used to choose a proper behavior, based on Sobot's internal state as well as stimulus information. When there is no command input from a user, various behaviors can be selected probabilistically by introducing a voting mechanism, where each behavior has its own voting value. The algorithm is described as follows: 1) Determine temporal voting vector, V_t using M and H, 2) Calculate voting vector V by masking V_t with attention command and emotion masks, 3) Calculate a behavior selection probability, $p(b)$, using V, 4) Select a proper behavior b by $p(b)$ among various behaviors.

Initially, the temporal voting vector is calculated from the motivation and homeostasis as follows:

$$\begin{aligned}\mathbf{V}_t^T &= \left(\mathbf{M}^T\mathbf{D}_M + \mathbf{H}^T\mathbf{D}_H\right) \\ &= [v_{t1}, v_{t2}, \cdots, v_{tn}]\end{aligned} \tag{1}$$

$$\mathbf{D}_M = \begin{pmatrix} d_{M11} & d_{M12} & \cdots & d_{M1n} \\ d_{M21} & d_{M22} & & \vdots \\ \vdots & & \ddots & \\ d_{Mx1} & & & d_{Mxn} \end{pmatrix}$$
$$\mathbf{D}_H = \begin{pmatrix} d_{H11} & d_{H12} & \cdots & d_{H1n} \\ d_{H21} & d_{H22} & & \vdots \\ \vdots & & \ddots & \\ d_{Hy1} & & & d_{Hyn} \end{pmatrix} \tag{2}$$

where n, x and y are the numbers of behaviors, motivations, and homeostases. v_{tk}, $k = 1, \cdots, n$, is the temporal voting value, D_M and D_H are weights connecting the motivation and homeostasis to behaviors, respectively.

As a next step, various maskings to the temporal voting vector, V_t are implemented considering emotion and external sensor information. Here, three kinds of masking are implemented to the temporal voting vector. These three kinds of maskings are 'masking for attention,' 'masking for command,' and 'masking for emotion.' The masking process is to select a more appropriate behavior such that it prevents Sobot from carrying out unusual behaviors. For example, a behavior when it recognizes a ball should be different from that when it recognizes a person. When Sobot does not see the ball, 'masking for attention' to the ball is carried out such that behaviors related to the ball are masked out and are not activated.

An attention masking matrix $\mathbf{Q}^a(S_a(t))$ is obtained by the attention symbol, $S_a(t)$. Each attention symbol has its own masking value and the matrix is defined as follows:

$$\mathbf{Q}^a(S_a(t)) = \begin{pmatrix} q_1^a(S_a(t)) & 0 & \cdots & 0 \\ 0 & q_2^a(S_a(t)) & & \vdots \\ \vdots & & \ddots & \\ 0 & & & q_n^a(S_a(t)) \end{pmatrix} \tag{3}$$

where n is a number of behaviors, $q^a(\cdot)$ is a masking value, and $0 \leq q^a(\cdot) \leq 1$. Similarly, command and emotion masking matrices are defined.

From these three masking matrices and the temporal voting vector, the behavior selector obtains a final voting vector as follows:

$$\begin{aligned}\mathbf{V}^T &= \mathbf{V}_{temp}^T \mathbf{Q}^a(a)\mathbf{Q}^v(c)\mathbf{Q}^e(e) \\ &= [v_1, v_2, \cdots, v_n]\end{aligned} \tag{4}$$

where v_k, $k = 1, 2, \cdots, n$, is the kth behavior's voting value.

Finally, the selection probability $p(b_i)$ of a behavior, b_i, $i = 1, 2, \cdots, n$, is calculated from the voting values as follows:

$$p(b_i) = \frac{v_i}{\sum_{k=1}^{n}(v_k)}. \tag{5}$$

By using the probability-based selection mechanism, the behavior selector can show diverse behaviors.

Even if a behavior is selected by both internal state and sensor information, there are still some limits on providing Sobot with natural behaviors. 'Inherent behavior selector' makes up for the weak points in the behavior selector. It imitates an animal's instinct. For instance, as soon as an obstacle like a wall or a cliff is found, it makes Sobot react to this situation immediately. Since it uses only sensory information directly, its decision making speed is faster than that of the behavior selector. The deterministic inherent behavior selector and the probabilistic behavior selector are complementary to each other for realizing a natural behavior. This means that it can help Sobot do right thing while carrying out various behaviors.

3.4 Motor module

The motor module incorporates an actuator to execute behaviors and present emotions subject to the situation.

3.5 Learning module

Learning module consists of preference learner and command learner. The former is to teach Sobot likes and dislikes for an object. If Sobot gets a reward or a penalty, the connected weights from the symbol to internal states are changed. On the other hand, the latter is to teach Sobot to do an appropriate behavior which a user wants Sobot to do.

The learning can be considered as adjusting weighting parameters between commands and behaviors; if Sobot does a proper behavior for a given command, the weight between the command and the behavior is strengthened, and others are weakened. However, there are usually tens of behaviors. Thus, the learning process requires lot of time. Also it may be difficult to expect a desired behavior for an ordered command. To solve these problems, analogous behaviors are grouped into a subset before learning. For instance, the set 'SIT' is composed of behaviors such as sit, crouch, and lie, and so on, as similar behaviors to 'sit.' If a proper behavior is carried out for a certain command, all the corresponding weights of the subset are strengthened and vice versa. The update law is as follows:

$$W_{ij}(t+1) = W_{ij}(t) + \rho R_i \tag{6}$$

$$R_i = \begin{cases} +C_r & \text{on reward} \\ -C_p & \text{on penalty} \end{cases}$$

where W_{ij} is a weight between the ith command and the jth behavior subset, ρ is an emotion parameter, R_i is for a weight change for reward or penalty, and C_r and C_p are positive constants. When Sobot receives a patting (hitting) through a tactile sensor or a praise (a scolding) through a sound sensor, the perception module translates it as a reward (penalty). Weight is increased on reward, and decreased on penalty as shown in (6). It should be noted that an emotion parameter, ρ is employed to consider the fact that learning rate depends on internal states. That is, learning speed is fast when happiness value is high and vice versa.

Although the learning has been done on a behavior subset level, considering the direct contribution of the selected behavior the command masking values are assigned differently as follows:

$$\begin{aligned} q_m^v(c_i) &= \alpha W_{ij} \\ q^v(c_i) &= \beta W_{ij} \end{aligned} \tag{7}$$

with $\alpha > \beta > 0$

where $q_m^v(c_i)$ is a masking value of a behavior, b_m carried out just now by the command, C_i and $q^v(c_i)$ indicates masking values of other behaviors in the same subset, B_i and α and β are positive constants. The command masking matrix is updated in proportion to weight values. A behavior activated just now and other behaviors in the same subset influence different weight changes by α and β. Since α is bigger than β, the activated behavior gets a larger weight value than others in the same subset.

4 Demonstrations

To demonstrate the usability of Rity for Ubibot, a Sobot, Rity is developed in a 3D virtual world. The following two demonstrations show seamless and omni-presence properties of Sobot.

4.1 Seamless integration of real and virtual worlds

This section will demonstrate how, in a virtual environment, Rity will continuously cooperate with the real world with the help of a USB camera. The face recognition system stored in a PC watches the neighboring environment through the USB camera and, when a human is detected, analyzes, recognizes and authenticates the face. The result is to be sent to Rity through the network. Sobot will then react to the vision input information as it would normally react using the virtual sensing information. If the human is Rity's master, Rity will tend to stare at the master and happily greet him/her.

Fig. 3, 4 and 5 are photographs of computer screens showing the virtual pet, Rity, in a virtual 3D environment. The small window at the bottom right of Fig. 3 shows the visual information in the form of a recognized face. A PCA method[17], which has been enhanced based on the evolutionary algorithm, was used for face detection. The window at the top right shows the graphical representation of the internal states of Rity.

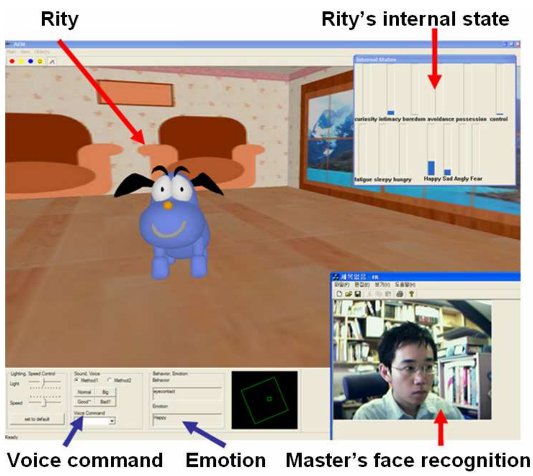


Figure 3: Seamless integration of real and virtual worlds

Fig. 4 shows an example, in which Rity recognizes its master. Rity then shows a happy look and welcomes him, with an increase of such internal states as curiosity, intimacy, and happiness.

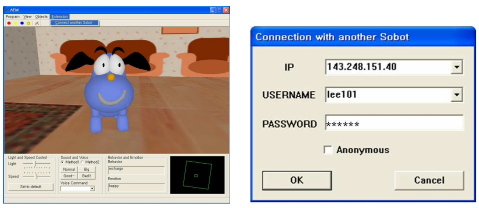
Figure 4: When Rity recognizes its master

In Fig. 5, when a human who is not the master appears, Rity ignores him/her. In this case, for example, the internal state keeps as it has been.

Figure 5: When Rity detects a stranger

4.2 Omni-present Sobot

This section discusses how Sobot can be connected and transmitted any time and at any place. Fig. 6 shows the interaction between Sobot A, owned by User A and Sobot B, owned by User B.

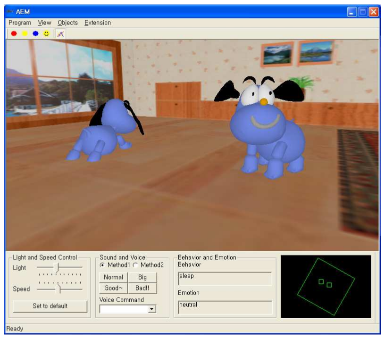


(a) (b)

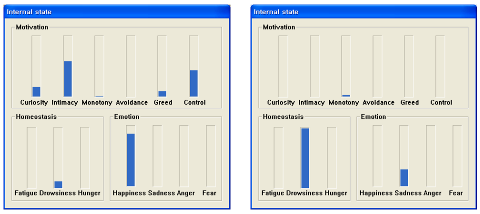
Figure 6: Omni-present Sobot (a) connection with another Sobot in a remote site (b) IP address of a Sobot in a remote site, username and password for certification

For example, Sobot A is implemented at a local site, connects to the network and then invites Sobot B, located at a remote site, into its local space. Both Sobots (A and B) should have their own individual IP addresses. The User B will type in the ID and password and the IP address of Sobot B in order to access the remote site. Once access is approved, Sobot B, carrying its native characteristics and behavior patterns, can enter the local environment of User A.

In Fig. 7, there are two Sobots in the local space. They look the same but have different characteristics. If the user gives the same stimulus to the two Sobots, for example, clicks once to pat or twice to hit, each Sobot will react differently because of their different characteristics. Fig. 7 shows the results of the experimentation after applying 10 instances of patting, or clicking, on both Sobots A and B. The figure shows the changes in internal states, facial expression and their behavior. As the amount of curiosity, intimacy and happiness increases, Sobot A starts moving around with a happy face, Fig. 7(a). On the other hand, in the case of Sobot B, the drowsiness increases making it sad and eventually sleepy. Fig. 7(b) and7(c) shows a comparison of the internal states of Sobot A and B.



(a)



(b) (c)

Figure 7: Omni-presence (a) Sobot A in a local site and Sobot B downloaded from a remote site (b) Internal state of Sobot A (c) Internal state of Sobot B

Sobot can be downloaded and sent regardless of whether the site is local or remote. This is made possible by defining a common platform of the 3D graphic environment along with sensors and behaviors.

5 Concluding Remarks

In this paper, as a third generation of robotics a ubiquitous robot, Ubibot, was introduced, which integrates three forms of robots: Sobot, Embot and Mobot. Sobots, which are software-based virtual robots in virtual environments, can traverse space through physical networks. Embots, the embedded robots, are implanted in the environment or embedded in Mobot, for sensing, detecting, recognizing, and verifying the objects or the situation. The processed information is to be transferred to Sobot or Mobot. Mobots provide integrated mobile services that Sobots and Embots cannot. Sobots and Embots can work individually or within Mobots.

Rity, a 3D character and a Sobot, was introduced and implemented using two scenarios to demonstrate the possibility of realizing Ubibot. The first scenario illustrated how Rity, with the support of Embot, could recognize its master and reacted properly. This was to show the seamless integration of real and virtual worlds. The second scenario demonstrated how Sobots could be transmitted through networks and be transposed into different locations. This was to demonstrate the omni-presence capability by using Sobot.

In the new ubiquitous era, our future world will be composed of millions of u-spaces, each of which will be closely connected through ubiquitous networks. In this u-space we can expect that Ubibot will help us whenever we click as Genie of the Aladdin Magic Lamp did.

Acknowledgments

This work was supported by the Ministry of information & Communications, Korea, under the Information Technology Research Center (ITRC) Support Program.

References

[1] Jong-Hwan Kim, “Ubiquitous Robot,” in *Proc.of Fuzzy Days International Conference*, Dortmund, Germany, September 2004, (Keynote Speech Paper).

[2] Mark Weiser, “The computer for the 21st century,” *Scientific American*, Vol. 265, No. 3, pp. 94-104, Sept. 1991.

[3] Mark Weiser, “Some computer science problems in ubiquitous computing,” *Communications of ACM*, Vol. 36, No.7, pp. 75-84, July 1993.

[4] Jong-Hwan Kim, “ IT-based UbiBot,” in *the Issue of the 13th of May, 2003, The Korea Electronic Times*, Special Theme Lecture Article, Seoul, Korea, May 2003.

[5] Y.-D. Kim, Y.-J. Kim, J.-H. Kim and J.-R. Lim, “Implementaton of Artificial Creature based on Interactive Learning,” in *Proc.of FIRA Robot World Congress*, Seoul, Korea, pp. 369-373, May 2002.

[6] C. Breazeal, “Function Meets Style: Insights From Emotion Theory Applied to HRI,” *IEEE Trans. on Systems, Man, and Cybernetics, Part C*, vol.32, no. 2, pp. 187-194, May 2004.

[7] H. Miwa, T. Umetsu, A. Takanishi, and H. Takanobu, “Robot personality based on the equation of emotion defined in the 3d mental space,” in *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 3, Seoul, Korea, pp. 2602-2607, May 2001.

[8] J. Bates, A.B. Loyall and W.S. Reilly, “Integrating Reactivity, Goals, and Emotion in a Broad Agent,” in *Proc. of 14th Ann. Conf. Cognitive Science Soc.*, Bloomington, IN, July 1992.

[9] M. Mateas, “An Oz-Centric Review of Interactive Drama and Believable Agents,” *AI Today: Recent Trends and Developments*, Lecture Notes in Artificial Intelligence no. 1600, pp. 297-328, Springer-Verlag, Berlin, 1999.

[10] C. Kline and B. Blumberg, “The Art and Science of Synthetic Character Design,” in *Proc. of the AISB 1999 Symp. on AI and Creativity in Entertainment and visual Art*, Edinburgh, Scotland, 1999.

[11] J.-D Velásquez, “An emotion-based approach to robotics,” in *Proc. of IEEE/RSJ Int. Conf. on Intellighent Robots and Systems*, vol. 1, Kyongju, Korea, Oct. 1999, pp. 235-240.

[12] N. Kubota, Y. Nojima, N. Baba, F. Kojima, and T. Fukuda, “Evolving pet robot with emotional model,” in *Proc. of IEEE Congress on Evolutionary Computation*, vol. 2, San Diego, CA, pp. 1231-1237, July 2000.

[13] R. C. Arkin, M. Fujita, T. Takagi and R. Hasehawa, “Ethological Modeling and Architecture for an Entertainment Robot,” in *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol.1, Seoul, Korea, May 2001, pp. 453-458.

[14] D. Isla, R. Burke, M. Downie, and B. Blumberg, “A Layered Brain Architecture for Synthetic Creatures,” in *Proc. of the Int. Joint Conf. on Artifical Intelligence*, Seattle, WA, Aug. 2001, pp. 1051-1058.

[15] S.-Y. Yoon, B. M. Blumberg, and G. E. Schneider, “Motivation driven learning for interactive synthetic characters,” in *Proc. of the fourth Int. Conf. on Autonomous Agents*, Barcelona, Spain, Jun. 2000, pp. 365-372.

[16] B. Kort, R. Reilly and R. D. Picard, “An Affective Model of Interplay Between Emotions and Learning: Reengineering Educational Pedagogy - building a Learning Companion,” in *Proc. of IEEE Int. Conf. on Advanced Learning Technologies*, Madison, WI, Aug. 2001, pp. 43-46.

[17] J.-S. Jang, K.-H. Han, and J.-H. Kim, “Face Detection using Quantum-inspired Evolutionary Algorithm,” *in Proc. of the Congress on Evolutionary Computation*, Portland, OR, Jun. 2004, pp. 2100-2107.