PEXIS: Probabilistic Experience Representation Based Adaptive Interaction System for Personal Robots

Tetsunari Inamura,1,2 Masayuki Inaba,2 and Hirochika Inoue2

1The Japan Science and Technology Corporation, CREST, Saitama Prefecture, 332-0012 Japan
2Department of Mechano-Informatics, Faculty of Engineering, The University of Tokyo, Tokyo 113-8656 Japan

SUMMARY

In this paper the authors focus on the interaction between users and personal robots that can move in a real environment. When the creation of robots that can perform in an ordinary home or office is considered, it is difficult to imagine beforehand what kind of environment will be used, and so the approach in which developers embed environmental knowledge and strategies for autonomous movement fails. Thus, the authors propose an approach in which knowledge of the environment and the knowledge needed to move are acquired after development by allowing the robot and the user who is using the robot to engage in dialogue, and representing experience statistically. The authors introduce PEXIS, an interaction system developed using this approach, and then describe the characteristics and utility of their system via examples such as learning to avoid objects in an office environment and adapting to the vocabulary expressions of a particular user. © 2004 Wiley Periodicals, Inc. Syst Comp Jpn, 35(6): 98–109, 2004; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/scj.10034

Key words: personal robot; human–robot interaction; Bayesian Network; adaptation; learning.

1. Introduction

In recent years there has been an increased focus on personal robots that assist with daily life as robots have entered the environments that people live in and have begun communicating with them. Robots that are presumed able to interact with users are expected to bring about a variety of breakthroughs as a result of making the most of dialogue with users. One aspect of this is represented by an effort to have robots accumulate dialogue with users as experience, then learn and adapt based on their memories and experiences. The environments in which a robot is used cannot be predicted, and so knowledge and autonomous behavior cannot be embedded beforehand. The approach of making use of dialogue experience is useful for this problem.

Here, the elements necessary for an interaction system for a personal robot are summarized.

First, as the word “personal” suggests, behavior and dialogue based on the living environment of the individual users and background knowledge about the user must be implemented. To put this differently, the robot should adapt to the individual user, and engage in behavior that feels progressively more familiar. In order to deal with “personal” information such as the user’s preferences or particular ways of using words, the system and the user should converse for a short period, and then the system can adapt based on experience.

Second, the robot does not have an infrastructure for behavior, and it must behave in an unknown environment.

© 2004 Wiley Periodicals, Inc.
2. Adaptive Interaction Based on Probabilistic Experience Representation

2.1. Using a Bayesian Network

When giving a robot that has no knowledge new information, a general instructional method [1] involves giving instructions that have a one-to-one correspondence with behavior to be performed based on environmental information and the robot’s internal state. If the environment changes and the acquired model cannot be used any more, instructions must be given again from the start. In addition, there is no guarantee that the instructions a person gives are complete, and a person may make a mistake. This results in a model that reflects the errors.

This seems to be a result of the relationship between environmental information that varies dynamically and the behavior to be performed being represented in a static fashion. In order to avoid this problem, a statistical model referred to as a Bayesian Network [2] is used as an approach to represent the results of dialogue in a probabilistic manner using statistics rather than directly storing the results. A Bayesian Network is an estimation model that represents the cause-and-effect relationships between several phenomena using probability. This method [7] is widely used in fields such as map acquisition for mobile robots [3, 4], computer error diagnosis [5], and interactive systems [6].

The advantages of this model are:

(1) An online decision-making model can be acquired using simple statistical calculations based on experience accumulated from interaction between the person and the robot.

(2) Because this model is more suitable for handling symbols than neural networks or other methods, the results of language interaction are more easily represented, and the relationships between phenomena can be explained using symbols.

(3) The relationship between the user and the robot and the relationship between the robot and its environment can be described at the same level of representation, and so a behavioral model that reflects the user’s tastes and preferences can be created.

(4) The robot can make decisions with some precision by taking advantage of its experience using only a portion of the amount of information that would be necessary to make a decision.

2.2. Definition of experience representation

A Bayesian Network is a type of directed acyclic graph, and consists of a network of parent nodes representing causes and child nodes representing results. Each node has several propositions assigned to it. The states, conditions, and behaviors of robots are represented as activated propositions for each node. We call this a propositional symbol. The activation of a proposition is represented probabilistically, and as a result each node has a stochastic variable. In specific terms, sensor information in the robot, behavior that the robot is to perform, and content provided by a person are assigned to nodes. The relationship among nodes is described using conditional probabilities with stochastic variables.

Let us assume that the stochastic variable for a particular node B with the propositional symbols $b_1, \ldots, b_n$ is $B = \{b_1, \ldots, b_n\}$. The set of parent nodes connected from above to this node is $A = \{A_1, \ldots, A_m\}$, and the space consisting of combinations of each value for the stochastic variables is $a_1, \ldots, a_i$. Then the relationship between nodes is represented by the following matrix.

$$CPT_{B|A} \overset{\text{def}}{=} P(B|A) = \begin{pmatrix} P(b_1|a_1) & \cdots & P(b_n|a_1) \\ \vdots & \ddots & \vdots \\ P(b_1|a_m) & \cdots & P(b_n|a_1) \end{pmatrix}$$

This is called a CPT (Conditional Probability Table). This CPT is calculated for all parent–child nodes.

2.3. Adaptation based on interactive dialogue

The CPT defined above is not fixed, but rather adapts to the user via interaction between the user and the robot. This adaptation can be broken down into five stages: observation phase, learning phase, reasoning phase, dialogue phase, and introspection phase.

(a) Observation phase

Fundamentally, the robot behaves in accordance with the user’s instructions. Within its behavior, the robot ob-
serves what states it encounters, what instructions it receives, and what sensor information it picks up. The observation results are converted to the propositional symbols described above, and then stored in a separate database for each individual user.

(b) Learning phase

A cumulative frequency index obtained from simple experiments is used to update the cause-and-effect (CPT) relationship between node A with stochastic variable \( A \) and node B with stochastic variable \( B \). Let \( N \) represent the number of experimental data points observed, and let \( N_j \) be the number of times the propositional symbol \( a_j \) for node A is observed. Also, let \( n_{ij} \) represent the number of times the propositional symbol \( b_i \) is observed for node B among the \( N_j \) observations. Then, each component \( p_{ij} \) for the cause-and-effect relationship \( CPT_{BA} \) found between nodes A and B is

\[
P_{ij} = P(B = b_i | A = a_j) = \frac{n_{ij}}{N_j} \tag{2}
\]

The a priori probability for the parent node is

\[
P(a_j) = P(A = a_j) = \frac{N_j}{N} \tag{3}
\]

The "number of times" is equivalent to the number of times indicated by the user, or the number of times obtained from sensor information. This calculation is performed for all links and between all nodes.

(c) Reasoning phase

When the robot engages in autonomous behavior, the probability that each command for a node (a behavioral node is the simplest) necessary for decision-making will be created is calculated, and the decision corresponding to the command with the highest probability is then made. In this instance, the system gathers the information for evidence and inputs it into a Bayesian Network so as to make the best choice.

Here, if the evidence data in the network is \( E \), then the probability for node \( X \) can be represented as

\[
P(X|E) = \alpha P(E_X|X)P(X|E_X^+) \tag{4}
\]

because of the effects of both \( E_X^+ \) from the parent node and the effects of \( E_X^- \) from the child node (refer to Ref. 8 for the representation method). Here, \( \alpha \) is the normalization coefficient. When the network structure includes overlapping trees, the parent node for \( X \) is \( U = \{ U_1, U_2, \ldots, U_m \} \), and the child node for \( X \) is \( Y = \{ Y_1, Y_2, \ldots, Y_n \} \). Thus,

\[
P(X|E_X^+) = \sum_u P(X|u) \prod_i P(u_i|E_u/i|X) \tag{5}
\]

\[
P(E_X^+|X) = \beta \prod_i \sum_{y_i} P(E_{Y_i}|y_i) \sum_{z_i} P(y_i|X, z_i)
\]

\[
\prod_j P(z_{ij}|E_{Z_{ij}}/y_i) \tag{6}
\]

Note that \( \beta \) is the normalization coefficient. \( Z_i = \{ z_{i1}, \ldots, z_{ij} \} \) refers to the nodes excluding \( X \) among the parent nodes of node \( Y_i \), and \( E_u/X \) refers to those excluding evidence transmitted from node \( X \) among the evidence related to node \( U_i \). Also, here \( P(X|u) \) corresponds to \( CPT_{MU} \).

Equations (5) and (6) both involve iterative calculations, with computation progressing as information is transmitted across the entire network structure [2]. When evidence is observed at a node in the network structure, the probability of the goal node can be calculated by transmitting it up to a node where the information is not observed. This calculation clearly involves a computational burden proportional to the number of nodes assuming that the network does not contain a loop structure. The probability \( P(X|E) \) for node \( X \) is represented using \( BEL(X) \), and is interpreted as the vector with probability values sorted for the propositional symbols belonging to nodes. This is referred to as the certainty factor for node \( X \), and is used in determining behavior and selecting dialogue strategies.

As was described above, after a certain amount of experience has been amassed, the certainty factor for the remaining nodes can be reasoned using only information obtained from a part of the network.

(d) Dialogue phase

Although problems do not occur in a robot that has undergone sufficient learning, when the amount of observation is minimal, the confidence value is unstable, and
erroneous decisions can be made about behavior. In such states, the robot should be able to receive instructions from the user while continuing its autonomous behavior. This is an approach which does not aim for complete autonomy from the outset, but rather accumulates autonomous behaviors in stages. In a state that combines autonomous and directed states, dialogue between the person and the robot is important in order to establish which has priority. Here, the certainty factor described above is used in order to control the dialogue with the user.

Four levels of autonomous behavior can be set based on how high the certainty factor is. For each level, two states, yes and no for whether the user's instructions match the robot's decision, can be set. As a result, eight states are defined, with dialogue strategies assigned to each. There are four types of dialogue strategies: "indicate a user error," "propose an alternate plan," "create a question," and "confirm the user's instructions." For instance, if the certainty factor for a propositional symbol reasoned by the robot is extremely high and the robot determines that the user has made an error, then the error will be pointed out. Natural language corresponding to each propositional symbol is defined, and natural utterances can be generated using these definitions. Specific examples will be given in Section 4.

(e) Introspection phase

A threshold value used for making values discrete must be determined in order to have the command accompanying the symbol correspond to the node a sensor or other continuous value in the robot. The determination of this threshold value should be carried out afterwards based on experience obtained via dialogue with the user. The developers should not determine it beforehand.

Thus, in order to determine the threshold value using the experience of dialogue with the user, a parameter to be referred to as a reasoning error must be introduced.

When dialogue has been exchanged in the past, the difference between the content of instructions from a person and the propositional symbol reasoned by the robot at that time is given attention as an index which represents error in the decision-making model. \( \theta \) will be the threshold value parameter, the certainty factor for a node which the robot reasoned at time \( t \) based on such conditions will be \( \text{BEL}_t(B, \theta) \), and the instruction given by the user regarding that node will be \( b_t \). Thus, the next vector

\[
T_t = \{T_{t1}, \ldots, T_{tn}\}
\]

\[
T_{tij} = \{\delta_{ij}\} \quad (i = 1, \ldots, n)
\]

is defined using the Kronecker's delta \( \delta_{ij} \). The error value reasoned at time \( t \) is expressed as

\[
e_t(\theta) = ||\text{BEL}_t(B, \theta) - T_t||
\]

the sum of the absolute value of the difference in each component of the two vectors. The average is taken for the entire history of instruction for this error value, and the reasoned error value for the decision-making model becomes

\[
E(\theta) = \frac{1}{N} \sum_{i}^N e_t(\theta)
\]

\( N \) is the total number of historical data points. When this value is small, it can be determined that the decision-making model closely reflects the instructions of the user.

This reasoned error value for the decision-making model is a function of \( \theta \). The reasoned error value should take the smallest \( \theta \) possible in the parameter space for the threshold value. Here, because the search space is extremely large, the search is performed using a genetic algorithm (GA) with the reasoned error value \( E(\theta) \) as an indicator of appropriateness.

To summarize the above, the system updates the states of a Bayesian Network while performing the iterative operations shown in Fig. 2.

3. PEXIS: The Human–Robot Interaction System

Based on the ideas in the previous section, the authors developed PEXIS (Probabilistic Experience representation
based human–robot Interaction System), a combined system for interactions between robots and users. PEXIS is run as a server, and the robot and user take the form of clients connected to this system, as shown in Fig. 3.

3.1. PEXIS API

In order to support the software development for the robot, the authors prepared the PEXIS API. This API is a collection of functions used for decision-making and dialogue control. It can be used with robot platforms for various OSs and development languages through the use of communications protocols via character strings. The following command functions are available.

PEXIS::MakeBayesianNetwork (BN)

This declares the use of a Bayesian Network. In PEXIS, several networks can be created at the same time. The network to be used is designated with BN as the ID.

PEXIS::InputEvidence (n, p, r)

This inputs the raw data r generated from observed data at node n, then inputs the propositional symbol p corresponding to it at the same time.

PEXIS::QueryReasoningResults (n)

This finds the probability, that is, the certainty factor, for each propositional symbol corresponding to the intended node n.

PEXIS::DecisionMaking (n)

This performs decision-making using the certainty factor for node n. When the probability for two symbol elements is the same, the dialogue behavior previously described is triggered.

PEXIS::StoreDataBase (BN, DB)

This stores the states of the observed data already obtained in the database DB.

PEXIS::CptRevisionOnline (BN, DB)

This updates the CPT including the Bayesian Network BN based on the database DB.

PEXIS::ThresholdRevisionOffline (BN, DB)

This adjusts the threshold value for each node in the Bayesian Network BN offline so that it matches the past experience DB as closely as possible, then adapts to the user.

First, the developer provides a Bayesian Network appropriate for the robot's task to the robot using the MakeBayesianNetwork command. Next, the user's instructions and the observed data are input as evidence data to the Bayesian Network using the InputEvidence command. During the period in which the user is providing instructions to the robot, learning proceeds through the use of StoreDataBase and CptRevisionOnline, and when the robot performs autonomously, it decides on behavioral strategies autonomously using the QueryReasoningResult and DecisionMaking commands, and engages in dialogue. A concrete example of the PEXIS API in use will be provided in the next section.

3.2. PEXIS software configuration

As is illustrated in Fig. 3, PEXIS is an interface positioned between robots and users. PEXIS can be connected simultaneously between several robots and several users, and is not dependent on the type of robot or the form of the information input by users. Users can provide instructions and ask questions through voice interaction, a keyboard, joystick, or GUI. PEXIS can be broadly broken down into four subsystems which create overall behavior while communicating with each other.

(1) Natural language processing subsystem

Here, text analysis is performed on natural language in Japanese. As the meaning of each word is found, database

Fig. 3. Software configuration of the PEXIS.
reference commands to respond to user's questions and commands to engage in reasoning are generated, and each subsystem is driven. The utterance being addressed is Japanese text written in the Roman alphabet. Text obtained from voice recognition or keyboard input is analyzed here. This subsystem was created under EusLisp [10], one form of object-oriented Lisp based on TQAS [9] by Tanaka and colleagues.

In addition, the description of the relationship between the propositional symbols allocated to each node and natural language is created in this subsystem.

(2) Database subsystem

Here, the results of dialogue exchanged between the user and the robot and sensor information is stored as time-series experiential data. Also, when the robot is offline and not engaged in performing behaviors, the accumulated experience is put through introspection, as described above in the "Introspection Phase." This allows a threshold value appropriate for a particular user to be determined.

(3) Learning/reasoning subsystem

This primarily supervises the Bayesian Network computations. During training, the cause-and-effect relationships between each node for sensors, behaviors, and concepts are learned using Eqs. (2) and (3), and then represented using CPT. During reasoning, the certainty factor is found from experiential data. A propositional symbol with a sufficiently high certainty factor is sent to the connected robot, and the robot is triggered to engage in autonomous behavior. In addition, when there are two highest certainty factors, or when decision-making is difficult, the robot is triggered to start a dialogue with the user.

(4) User interface

This is prepared for several users using PEXIS. Linguistic information input from the voice recognition system is sent to PEXIS, and the utterances sent from PEXIS are displayed for the user using voice synthesis. In addition, functions for a joystick and GUI are also available. Sockets are used for communications with PEXIS. If PEXIS is used with a wireless terminal or wireless LAN, the user can engage in dialogue with the robot regardless of location.

4. A Practical Example: Acquisition of Behavior Strategies Based on Dialogue

4.1. Use in an obstacle avoidance task

First, the authors explain an example of acquiring behavior strategies based on dialogue between the user and the robot. An obstacle avoidance task was selected for a mobile robot as an example of autonomous behavior. This is a canonical task, and there is considerable research on training methods using neural networks [11] and training methods using reinforced learning [12]. However, there have been virtually no reports on examples of acquiring behavior through interaction between a person and a robot. Although Ref. 13 presents an example of acquiring behavior based on dialogue, this was limited to instructions for a sequence of behaviors.

The knowledge the robot initially has consists of two nodes: a sensor node $S = \{\text{Near, Middle, Far}\}$ and a movement node $B = \{\text{Forward, Left, Right, Stop}\}$, that is, only the Bayesian Network shown in Fig. 4. The robot has no mapping for sensors and motor commands, in other words no strategy for determining behavior.

In this state, the robot cannot take any action at all. As a result, at the initial stage, the user must give the robot instructions. The robot used for this experiment is a robot in a virtual environment as shown on the left in Fig. 5. It has eight distance sensors with values from 0 to 255, and has the task of going from Start to Goal while avoiding obstacles in the corridor-like environment shown in Fig. 6. In addition, the robot is connected to a user interface terminal as shown on the right of Fig. 5, and can be
controlled remotely by joystick guidance or voice instructions. Guidance from the person is input to the movement node using the InputEvidence command.

When instruction data is received, the robot starts to accumulate the data in the experiential database using the StoreDatabase command along with the sensor data at that instant. Then the conditional probabilities between the sensor node and movement node are updated online using the CPTRevisionOnline command. As a result, the robot can engage in a certain amount of decision-making after being guided a little bit. In this instance, behavior is selected and dialogue strategy is determined using reasoning output via the Bayesian Network, in other words the certainty factor for the movement nodes (the probability for go forward, turn left, turn right). Table 1 lists these situations.

Figure 7 shows how the certainty factor rises as the number of trials increases, and Table 2 shows an example of dialogue during the task. In this experiment, in the first trial, instructions and guidance were given entirely using the joystick. From the second trial onward, autonomous movement began. In Fig. 7 (1), the probabilities for go forward and turn left are close, and so confusion clearly results. From the third trial onward, appropriate behavior is clearly selected in every scene. In addition, the authors confirmed that similar behavior can be acquired in a real robot [14].

4.2. Individual adaptation during the introspection phase

In the obstacle avoidance task, because the stochastic variable for the sensor nodes is $S = \{Near, Middle, Far\}$, two threshold values used to make the sensor's continuous values discrete are needed. Updating in the "Introspection Phase" is performed in order to determine these threshold values as appropriate for an individual.

Guidance in two forms is performed: (1) guidance to safely and rapidly avoid obstacles when they are far away; and (2) guidance to select the most direct path and barely avoid obstacles. When the threshold values are determined beforehand, this difference in guidance cannot be reflected. However, movement that reflects the user's guidance, as shown in Fig. 8, can be learned using the ThresholdRevisionOffline command. In this instance, the threshold values start from a state in which all nodes have the same value. After five trials using (1) and (2), the threshold values in Fig. 9 for the sensor nodes are obtained. A division of labor can be seen for the central sensors and the edge sensors. Given this, it is clear that appropriate threshold values can be obtained via dialogue with the user.

5. Practical Example: Acquiring Vocabulary Adapted to a User

When considering a robot that operates in the real world, often objects are referred to using language such as

<table>
<thead>
<tr>
<th>The highest certainty factor &amp; Autonomy Level</th>
<th>User’s instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>80−100 [%] Autonomy Level 4</td>
<td>Point out the user’s mistake</td>
</tr>
<tr>
<td>65−79 [%] Autonomy Level 3</td>
<td>Make a suggestion of the inferred options</td>
</tr>
<tr>
<td>50−64 [%] Autonomy Level 2</td>
<td>Obey user’s instructions while showing the inferred options</td>
</tr>
<tr>
<td>0−49 [%] Autonomy Level 1</td>
<td>Obey user’s instructions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Difference Case</th>
<th>No instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute inferred options</td>
<td>Execute reasoning results with making suggestions as the inferred options</td>
</tr>
<tr>
<td>Execute reasoning results while showing inferred options</td>
<td>Wait for user’s operations while showing inferred options</td>
</tr>
</tbody>
</table>
Fig. 7. Development of learning in a virtual environment.

Table 2. An example of dialogue during the avoidance task

Robot: "Please tell me what I should do."
User: (Instructions using the joystick)
Robot: "Probably forward, but turn left."
Robot: "Probably turn left, but forward."
User: "A little to the left."
Robot: "Really turn to the left?"
User: "Yes."
Robot: "Understood."

"take the red book on the desk." When having a dialogue using the names of colors, for instance, the way people express a color may differ for the same color. In this section, PEXIS is used for this kind of problem. The authors describe an example of learning the relationship between the names of colors and color parameters in image processing via dialogue after the robot is created.

The authors used their system for a "visual search task" in which a robot with an upper body similar to a human (shown in Fig. 10) is to pick up an object identified by a user from among several objects on a desk. The visual characteristics parameters shown in Table 3 and the nodes for names people use to express these parameters were prepared for this task, and the network shown in Fig. 11 was set up. The intended object was identified through the dialogue shown in Table 4. The method for creating the particular dialogue is addressed in Refs. 15 and 16 in the interest of saving space here. The acquisition and adaptation to the relationships between color names and visual parameters are described here.

Fig. 8. A result of adaptation for two users.

Fig. 9. A result of threshold adaptation in the sensor nodes.

Fig. 10. An instruction experiment in an ambiguous situation.
Table 3. Random variables used in the visual search task

<table>
<thead>
<tr>
<th>Name of nodes</th>
<th>Random Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color-Parameter</td>
<td>$CP = {cp_0, cp_1, \ldots}$</td>
</tr>
<tr>
<td>Color-Name</td>
<td>$CN = {Red, Blue, \ldots}$</td>
</tr>
<tr>
<td>Location-Parameter</td>
<td>$LP = {Lp_0, Lp_1, \ldots}$</td>
</tr>
<tr>
<td>Location-Name</td>
<td>$LN = {Near, Middle, Far}$</td>
</tr>
<tr>
<td>Size-Parameter</td>
<td>$SP = {Sp_0, Sp_1, \ldots}$</td>
</tr>
<tr>
<td>Size-Name</td>
<td>$SN = {Small, Middle, Large}$</td>
</tr>
</tbody>
</table>

When the user expresses something like "take the red object," the expression used by the user is input into the Color-Name (CN) node. The name of the color used by the user is taken as evidence, and the visual parameters (Hue and Saturation) for the object designated by the user are acquired using introspection with respect to the Color-Parameter (CP) node which expresses the visual parameters. As a result, the object which most likely was indicated by the user is selected from among several objects. When the object being referred to is identified, the visual parameters obtained through image processing are input to the CP node, and then stored as experiential data with the expressions used by the user. CPT training between the stochastic variables $CP$ and $CN$ can be performed using this database.

In addition, in order to adapt to individual color expressions, the threshold values are updated using the ThresholdRevisionOffline command. What is revised here is the threshold values for the CP nodes shown at the bottom of Fig. 11. Because the space defined for Hue and Saturation is broken down in the areas $\{cp_0, \ldots, cp_{12}\}$ using 13 threshold values, adaptation to a user can be accomplished with $\theta = \{\theta_1, \ldots, \theta_{13}\}$.

Subjects interacted with the robot under good conditions and using color names freely with the network in an initial state in which $CN = \{Red, Blue, Yellow, Green\}$ and the CP node was broken down into 12 equal segments with respect to Hue. When the user used a new name for a particular color, this was handled by adding a new propositional symbol to the CN node. Figure 12 shows the relationship between the names of colors and physical parameters after roughly 50 interactions, and Fig. 13 shows the results of visual parameter thresholds corresponding to users.

Fig. 11. A Bayesian Network used in the visual search task.
Based on these figures, not only are the color names in the Hue parameter space not distributed uniformly, but also there are regions in which the same color name is used to represent slightly different Hue values, and regions in which the name changes for tiny differences in Hue. Given this, the authors were able to confirm that they had come up with a method to represent colors for individuals.

Here, although color representation was explained, if representations of size and position, such as "large" and "right," were used, the same kind of adaptation would result for the center and right nodes in Fig. 3.

6. Related Research

Okada and colleagues have proposed a learning system which varies behavior based on environmental conditions even when instructions are given in vague natural language to a mobile robot [17]. However, this system must regularly receive instructional information from a user, and so cannot be said to satisfy the conditions of autonomous behavior for a personal robot.

Nagao [18] has suggested the concept of a real-world object-oriented interface, and has pointed out the importance of an interaction system which combines users and real environments. The same can also be said for personal robots. In Nagao's system, the agent is a portable terminal carried in the user's hand. When a robot that moves autonomously is considered, the structure of the interaction changes significantly. This system includes elements necessary for the personal robot, which is equivalent to the agent, to move through the real world autonomously.

Matsui and colleagues [4, 19] have created an office service robot that they call Jijo-2. This robot also has dialogue and learning features, but there has been no discussion of adaptation to a particular user. The PEXIS system described in this paper can move from an office environment to a home environment, and can be used as a system which adapts to the user.

The conventional "learn by experience" framework has often dealt with experience between computers and users [20] and experience between robots and environments [12]. Within the framework of a personal robot, a model which addresses the robot, user, and environment at the same time is needed. The system described in this paper can address these three elements using the symbol representation in a Bayesian Network, and so is a method that is more applicable to personal robots.

7. Conclusion

The authors represented experience statistically in order to combine the functions needed for a personal robot, such as acquiring experience, learning, and knowledge, while interacting with a user in the real world, and then described a system which can interact adaptively using a
Bayesian Network. This system can be run via a server on the Internet, and so has a broad range of applications, including robots which perform services for several users, and behavior learning using remote control in networked robots.

Important issues include creating a method that can automatically generate a Bayesian Network for each task, and improving efficiency so that the maximum amount of learning is accomplished with the smallest number of interactions. As for the former issue, the amount of shared information among phenomena will be useful as an approach for creating networks automatically. As for the latter issue, the authors are attempting to represent important experience and experience that can be overlooked by weighting experience based on circumstance, and then to provide efficient training [14].

REFERENCES

Tetsunari Inamura (member) received his B.E. degree from the Department of Mechano-Informatics of the University of Tokyo in 1995 and D.Eng. degree in 2000. He was with the Japan Science and Technology Corporation, CREST, from 2000 to 2003. He is now a lecturer at the University of Tokyo, and is pursuing research related to human–robot interaction based on shared experiences, development of brain-like information processing mechanisms and stochastic systems. In 2003, he received the Best Conference Paper Award from the Japanese Society for Artificial Intelligence and the Funai Information Science Incentive Award from the Funai Information Science Foundation. He is a member of the Robotics Society of Japan, Japanese Society for Artificial Intelligence, and IEEE. He is also an editorial board member of the *Journal of the Robotics Society of Japan*.

Masayuki Inaba received his B.E. degree from the Department of Mechanical Engineering of the University of Tokyo in 1981 and D.Eng. degree in 1986. He is now an associate professor. His research interests include vision-based robotics, remote-brained robotics, robot system architecture, and developmental robot behaviors. He received an Outstanding Paper Award from the Robotics Society of Japan in 1987, Technical Paper Award from the Society of Instrument and Control Engineers of Japan in 1988, JIRA Award in 1994, and ROBOMECH Award from JSME in 1994. He is a member of the Robotics Society of Japan, Japan Society of Mechanical Engineers, Society of Instrument and Control Engineers, Information Processing Society of Japan, Japan Society for Artificial Intelligence, and IEEE.

Hirochika Inoue received his B.E., M.E., and Ph.D. degrees in mechanical engineering from the University of Tokyo in 1965, 1967, and 1970. He was with the robotics research division at the Electrotechnical Laboratory, MITI Japan, from 1970 to 1978. He was appointed an associate professor at the University of Tokyo in 1978, and has been a professor in the Department of Mechano-Informatics since 1984. Beginning in 1965, he has been engaged in robotics research and education, and he pioneered several important fields of robotics such as bilateral control of robot arm, visual guidance of robot motion, development of high-speed robot vision, view and visibility of environment, learning by seeing, and so on. For those accomplishments he received seven research paper awards from a variety of academic societies. His research interest covers almost all aspects of robotics, mechanical design, force control, vision-based robotics, language, planning, and system integration. He is a member of IEEE, ACM, JSME, Robotics Society of Japan, and Society of Instrument and Control Engineers, among others. He is also an editorial board member of the *International Journal of Robotics Research*, the *International Journal of Robotics Systems*, and others.