Behavior-Based Early Language Development on a Humanoid Robot

Paulina Varshavskaya (Varchavskaia)
Artificial Intelligence Lab, MIT
NE43-937, 200 Technology Square
Cambridge, MA 02139 US
paulina@ai.mit.edu

Abstract

We are exploring the idea that early language acquisition could be better modelled on an artificial creature by considering the pragmatic aspect of natural language and of its development in human infants. We have implemented a system of vocal behaviors on Kismet in which "words" or concepts are behaviors in a competitive hierarchy. This paper reports on the framework, the vocal system's architecture and algorithms, and some preliminary results from vocal label learning and concept formation.

1. Introduction

We are exploring the idea that early language acquisition could be better modelled on an artificial creature by considering the pragmatic aspect of natural language and its development in human infants. We believe that this will contribute to a solution to the "Grounding Problem" (Harnad, 1990) by providing a new level of grounding in intentions and function.

The pragmatic approach to language acquisition is to consider first of all the intentions of a speech act. Language is not viewed as a denotational symbolic system for reference to objects and relationships between them, as much as a tool for communicating intentions. The utterance is a way to manipulate the environment through the beliefs and actions of others.

We develop a system of vocal behaviors for the robot Kismet (Breazeal, 2000) which exemplifies the approach we believe should be taken to natural language acquisition by machines. The robot's youthful appearance dictates the sort of interaction that humans will have with it: scenarios of social scaffolding similar to the kinds of interactions that teachers have with human infants.

One step forward is to enable the robot to use the scaffolded environment to its advantage in order to learn to perform tasks and behave appropriately. Learning to communicate with the teachers using a shared semantic basis is one aspect of learning to behave in the world and manipulate it. We augment the existent motivational and behavioral systems of the robot with a set of vocal behaviors, regulatory drives, and learning algorithms, which together constitute Kismet's Protolanguage Module.

In what follows, we first take a look at previous work in robotics in Section 2., and in human language development in Section 3. Then we present the architecture of Kismet's protolanguage module in Section 4. and the algorithms in Section 5. Some preliminary results from experiments with the new system can be found in Section 6., and a discussion in Section 7.

2. Previous Work

The current work was inspired by and built upon results and ideas in robotic language acquisition and adaptable behavior-based robotic architectures.

2.1 Robots acquiring natural language

(Roy, 1999) and (Oates et al., 2000), approached the problem of acquisition of natural categories and labels by robots from the point of view of perceptual grounding. The robot analyzes the visual scene and the speech stream into segments, the best correlation between which will form a perceptual concept-label pair which is acquired by the robot, as for example in the development of CELL (Roy, 1999). CELL is embodied in an active vision camera and acquires lexical units from the following scenario: a human teacher places an object in front of the robot and describes it. The visual system extracts color and shape properties of the object, and CELL learns on-line a lexicon of color and shape terms grounded in the representations of objects. The terms learned need not be pertaining to color or shape exclusively - CELL has the potential to learn any words. Associations between linguistic and contextual channels are chosen on the basis of maximum mutual information.

2.2 Robots creating linguistic systems

Others have applied the principle of grounding word semantics in perceptual inputs of robots to acquisition and evolution of artificial languages, e.g. in (Steels, 1999) and (Steels and Kaplan, 2001). Those languages are used by populations of software agents which teleport themselves into active vision heads so they may perceive the scenes they describe to each other. These experiments are demonstrations of a computational model of language evolution. The resulting languages are not natural; they are constructed specifically and exclusively for the purpose of playing the naming game and not intended for interaction with humans. Therefore, a human interacting with an agent from the experiment must learn the new language in order to engage in communication, which will be restricted by the rules of the game.

Also interesting is the work of Holly Yanco (Yanco, 1994), where a team of robots must collectively perform an atomic task by following a designated leader (human or robot). Based on certain assumptions about the task and the environment, e.g., perfect communication, no inference of goals, and the simplicity of tasks, the robots create an artificial language to assist their collaborative effort. Again, the development of this language reflects its very specific purpose, which is not necessarily human-robot interaction.

2.3 Adaptable behavior-based architectures

We combine the task of word and concept acquisition with the behavior-based approach to robotics (Brooks, 1986) by conceiving of a concept as a kind of behavior, a process in a competitive hierarchy. However, traditionally behavior-based systems have fixed architectures of behaviors designed in advance for a specific purpose. Our purpose, on the contrary, is to build and modify an architecture of protoverbal behaviors at runtime, as the robot learns to extract them from experience. We therefore take an approach similar to that of certain recent expansions in behavior-based robotics, which have included adaptive architectures. For example, in (Nicolescu and Matarić, 2001) scenarios are described, in which humans act as teachers and collaborators for robots. Robots learn to combine existing behaviors into control networks from human demonstrations of tasks.

The work reported here builds on the Lateral behavior architecture (Fitzpatrick, 1997), which was extended and modified to enable adaptable behavioral architectures in (Varchavskaia, 2002).

2.4 Novel approach of reported work

The problem we are attempting to address in the current work is that of demonstrating the development of a communicative system by an artificial creature in a social environment populated by benevolent humans. This communicative system finds expression in strings of natural language. However, the emphasis of the current approach is not on learning an extensive vocabulary or a verisimilar grammar, but on acquiring words with functional meanings. Indeed, the questions of grammar development are not addressed here at all, hence we refer to the task as “protolanguage”, meaning the presyntactic stage in language development. Reported here are very preliminary results obtained by taking this pragmatic approach to language acquisition. The main novelty of this research is the framework proposed here, in which concepts have procedural as well as declarative meaning. Concepts (i.e., words with meanings) are processes in a concurrent “mental” architecture which compete to be expressed by the creature. To the best of our knowledge, this approach has not yet been explored in machine learning of natural language.

This work was most influenced and inspired by certain results in developmental linguistics, to which we now turn.

3. Development of Meaning and Language in Humans

Human infants are surprisingly adept at learning about the structure of the environment, how to behave in it, and how to express themselves and understand others all at once, in the space of a few years. Bloom (Bloom, 2000) has shown that children are good at innately facilitated learning of socially transmitted information. This means that one of the most important attributes of the spoken language in the infant’s environment is that it fulfills a social function, to which the infant is sensitive and which enables her to acquire a new word or concept after a very limited number of examples, perhaps even one (see also (Pinker, 1999) for overview of experiments suggesting extremely fast word learning by young children and infants).

In a seminal work (Halliday, 1975), most inspirational for the design of Kismet’s Protolanguage Module, Halliday makes the very important distinction between what he calls the *mathetic* and the *pragmatic* functions of human natural language. The mathetic function is to provide an encoding of information channelled through speech or text. When designing robots who develop their abilities in a manner similar to humans, we cannot simply focus on the mathetic function of language and neglect its pragmatic aspect, which is to provide the speaker with a means of manipulating the behavior of other humans. When one person makes a speech act, in addition to communicating a piece of information, that person may have intentions to change the state of the world and especially the behavior of those who hear the utterance, in a particular way. “It has started to rain outside” often really means something like “Close the window, please”. The speaker expects the hearer to react in a certain way as a result of hearing the utterance, which then becomes the speaker’s tool for manipulation of others and of her surroundings.

This pragmatic view of the function of language is extremely important in trying to explain, or devise, an ability for early language acquisition, because infants and young children specifically learn to use speech as a tool. Halliday identifies three main stages of linguistic development: (I) the child’s initial closed proto-linguistic system, (II) the transitional stage to that of adult language, and (III) the learning of the adult language. In the first stage, the child has a finite number of meanings to convey and to that effect uses self-generated labels that may or may not resemble adult words for similar occasions. Halliday posits six initial functions of a developing proto-linguistic system that may be expressed in the first stage:

1. **Instrumental** - satisfying the child’s needs
2. **Regulatory** - controlling the behavior of others
3. **Interactional** - engaging in a social situation
4. **Personal** - asserting own unique self
5. **Heuristic** - exploring the environment
6. **Imaginative** - pretending and playing

These six functions of the child’s phase I protolanguage seem to develop in that sequence and represent the child’s growing cognitive ability and awareness. They also present a great starting point and timeline for an artificial system that would acquire a natural language in a way similar to human children. The infant’s protolanguage during Phase I is finite and formulaic (see also (Wray, 2000) for a discussion of formulaic systems in the evolution of language), as will be the first part of Kismet’s language development module (see Table 1).

Note that we make no claims of faithfully following the developmental schedule of human infants. Neither do we achieve a complexity of development that would approach that of an infant. The preceding summary shows the main source of inspiration for our pragmatics-based approach to language acquisition by an artificial creature, designed to exhibit certain properties of human infants. As can be seen in the next two sections, learning in the artificial system results from similar scenarios of, e.g., frustration of a goal-directed behavior and an intuitive drive for vocalization.

4. Kismet’s Protolanguage Module

In order to achieve communication between humans and a sociable robotic creature, words must be a tool used by the robot to manipulate its physical and social world and they must be interpreted by humans as having such a pragmatic functional meaning. In Kismet’s case, it will start with proto-language and proto-verbal behaviors. The “proto” terms refer to the pre-grammatical early stage of development.

4.1 The robotic platform

Kismet is an expressive robotic head, designed to have a youthful appearance and perceptual and motor capabilities tuned to human communication channels. The robot receives visual input from four color CCD cameras and auditory input from a microphone. It performs motor acts such as vocalizations, facial expressions, posture changes, as well as gaze direction and head orientation.

Kismet’s control architectures run on a complex network of processors in real time (approaching 30 Hz for visual signals, and 8 kHz sample rate with frame windows of 10 ms for auditory signals), with minimal latencies (less than 500 ms). Low-level visual processing and eye/neck motor control is performed by 12 networked 400 MHz PCs running QNX. The high-level perceptual system, the motivation and behavior systems, the motor skill system and the face motor control run on four Motorola 68332 microprocessors running L, a multi-threaded Lisp developed in our lab. Expressive speech synthesis and vocal affect recognition execute on a dual 450 MHz PC running NT, and the speech recognition system (ViaVoice) and protolanguage module run on two 500 MHz PCs running Linux.

Although not a mobile robot and with all computation off-board, Kismet is an autonomous agent in that it pursues its own agenda by engaging in specific behaviors which are tuned to the satisfaction of its own “goals” and “desires”. The motivation is provided by a set of internal homeostatic variables called “drives”, such as the level of engagement with the environment or the intensity of social play, which must be maintained within certain normal bounds in order for Kismet’s system to be at equilibrium.

“Emotions” constitute another facet of Kismet’s motivational system. The robot’s emotional state is modelled, after Ekman, cited in (Breazeal, 2000), as a point in three-dimensional space, where the axes represent arousal, valence, and stance. The choice of emotion depends on simple appraisals of the perceptual stimuli. The robot has a 15 DoF face that mirrors its internal “emotional” state expressively.

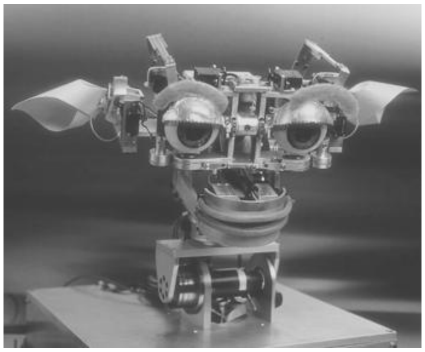


Figure 1: Kismet: the robotic head.

The behavior system provides structure and an arbiter for the robot's multiple behaviors. The latter are all self-interested mechanisms which compete with each other to be active at any one time. The activation of a particular behavior will depend on the current state of the robot's motivational system, as well as considerations of coherency, persistence and opportunism.

There are four lip actuators, and a single DoF jaw that together work to lip-synch to the speech produced by a synthesizer. The synthesizer software is DECTalk v4.5, based on the Klatt synthesizer, cited in (Breazeal, 2000), which models the physiological characteristics of the human articulatory tract. This enables the robot to speak in a youthful tone of voice, and to vary the parameters of the synthesizer to account for variation in its emotional state.

4.2 Proto-verbal behaviors

We have designed and implemented a new vocal behavior system on Kismet which we call protoverbal for two reasons. On the one hand, the behavior exhibited by the system, if observed in a human infant, would be called a precursor to language development. The goal of the system is to produce the kind of vocal output that a prelinguistic infant may produce in the age range of 10-12 months, namely emotive grunts, canonical babblings, and a formulaic protolanguage (see Table 1) similar to that described in (Halliday, 1975). On the other hand, we believe that this foundation of vocal behaviors can serve as the pragmatic basis for more sophisticated natural language acquisition by the robot.

The system consists of two new drives (the *Speech* and *Exploration* drives), and an architecture of vocal behaviors shown in Figure 2. The new system is based on the following components:

- **Releasers** are global variables which respond to certain conditions in the environment, as reported by the robot's perceptual system, or in the internal state of the robot

Emotion	Behavior	Proto-linguistic Function
anger, frustration	complain	regulatory
disgust	withdraw	instrumental or regulatory
fear, distress	escape	–
calm	engage	interactional
joy	display pleasure	personal or interactional
sorrow	display sorrow	regulatory or personal
surprise	startle response	–
boredom	seek	–

Table 1: Correspondence between Kismet's nonverbal behaviors and proto-linguistic functions (Halliday, 1975). In some fields, – indicates that there is no clear correspondence. In this case, the grunting behaviors may be active.

- **Drives** are global objects with a changing level of activation, which regulate the likelihood of activation of certain behaviors, and are reset when these behaviors achieve their goals

- **Vocal Behaviors** are behavior objects which consist of **receptors**, **gains**, **elicitors** which behave like complex releasers, a level of **activation**, a **state machine**, a **vocal label**, a measure of **confidence**, and **output ports** for propagation of their activation and label. Details of a base behavior are shown in Figure 3.

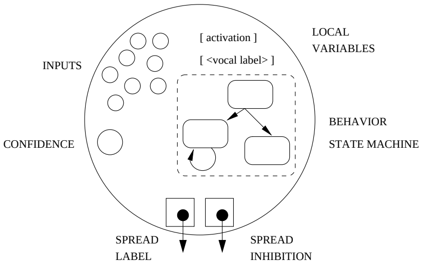


Figure 3: Representation of a single protoverbal behavior.

The design of protoverbal behaviors presented here fits the description of a behavior as a a self-contained, self-interested and goal directed entity. All of them compete to establish which mode of vocal expression the robot is to engage in, and what phonemic string it is to produce.

- **Self-interest**. Each vocal behavior computes its activation locally and attempts to overwrite its connections to other behaviors and to the `Say This` buffer containing the string that the robot

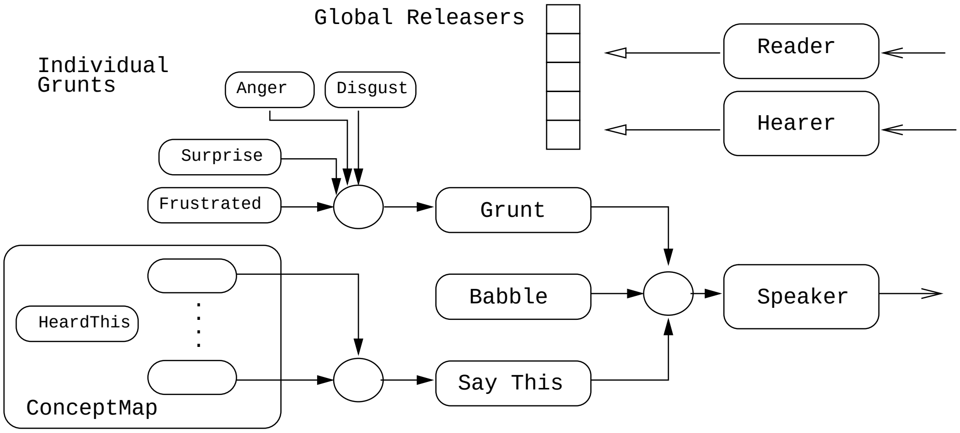


Figure 2: Overall architecture of Kismet's protoverbal behaviors, where rounded boxes represent instances of behaviors and circles represent connections between behaviors. Connections between `HeardThis` and individual `Concepts` are not shown for clarity (see section 5.1

will say next. A single vocal behavior does not depend on the operation of others. Behaviors may communicate indirectly by receiving each other's confidence signal as input.

- **Goal-directedness**. The goal of each vocal behavior is to assuage the robot's *Speech* drive by making it say something. Different kinds of vocalizations may satisfy different drives, e.g., the *Exploration* drive grows when there are no people present in the robot's environment, and is satisfied by canonical babbling.

- **Competition**. The competition between vocal behaviors is regulated by a priority scheme in the Lateral architecture (Fitzpatrick, 1997). Each behavior assigns its own priority locally to its computed activation level. Implementation details can be found in (Varchavskaia, 2002); they correspond closely to the original implementation of behavior activation on Kismet.

The system consists of grunting behaviors, a single canonical babbling behavior, and a number of concept behaviors (this number may grow or shrink at runtime). All of the above execute and interact in the overall architectural framework shown in Figure 2. There are two specialized behaviors, `Reader` and `Hearer`, which interface with Kismet's perceptual system and procure global releasers for vocal behaviors. A single `Speaker` behavior is responsible for sending a speech request over to the robot.

All of the protoverbal behaviors (rounded boxes on the figure) have access to the global releasers and the currently heard string of phonemes.

The nature of the speech request is determined by competition among individual protoverbal behaviors implemented through the Lateral priority scheme. To that effect, relevant behaviors write their vocal labels to connection objects, shown as small circles on the figure. The behavior with the highest activation, and therefore the highest priority, will succeed in overwriting all other request strings with its own, which `Speaker` will end up passing on. A speech request may be a request for a grunt, a babble, or a "word" - i.e., a phonemic string that is attached to one of the concept behaviors. Competition happens in two stages. First, the most active grunt writes its request to the `Grunt` buffer and the most active concept writes its label to the `Say This` buffer. Then, the most active of the three types of request buffers writes its output to `Speaker`. Any of these behaviors only produce output when activation is above a threshold (determined empirically), so some of the time, the Protolanguage Module does not produce an output, and the robot remains silent.

The vocal behaviors are influenced by data on the robot's current perceptual, emotional, and behavioral state. Figure 4 represents the way data and control flow between existing software components of Kismet's architecture and the vocal behaviors developed here.

The Perception, Behavior, and Motor Systems communicate the current values of `Simple Releasers`, implemented as variables of global scope, to which any component of `Vocal Behaviors` has access. `Complex Releasers` are computed by combining information from these and also become inputs to the new `Vocal Behaviors`. Finally, the outputs of the system are written directly to the speech stream, overwriting any existing value with the one determined by `Vocal Behaviors` and requesting a new speech act.

The entire protoverbal system shown on the right of figure 4 includes the implementation of algorithms for concept and vocal label acquisition and updates.

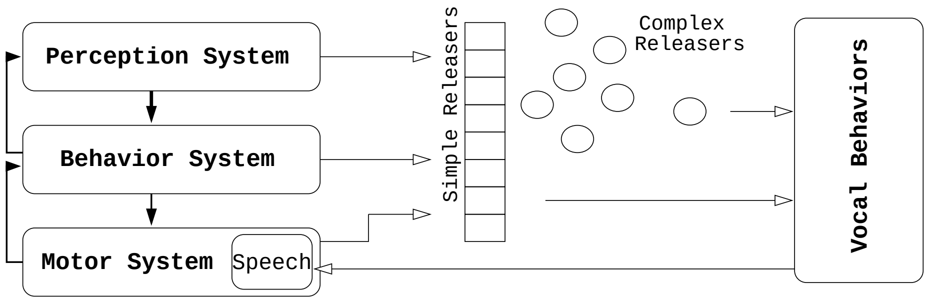


Figure 4: Control and data flow between vocal and other behaviors in Kismet's architecture. Dark arrows indicate combined control and data flow between old components of the architecture. Light arrows follow the data flows only.

5. Structures and Algorithms for Protolanguage Development

The development of the robot's formulaic proto-lexicon hinges on two mechanisms: one for the update of the vocal label of an existing concept (or "word"), the other one for the creation and management of novel concepts.

5.1 *Acquisition of labels*

The `Concept` class implements the functionality of a single vocal behavior. Its overall architecture was briefly explained in Figure 3, and its state machine is presented in more detail in figure 5. After initialization, the behavior can be in one of the seven states. On the figure, two non-default transitions are shown from the `Decide` state: to `Activate1` and `Hear` states. They are determined as follows: after the `OutputLabel` state has executed, the transition goes to `Activate1` unless the behavior receives a signal that there is a new speech input, in which case the transition goes to `Hear`.

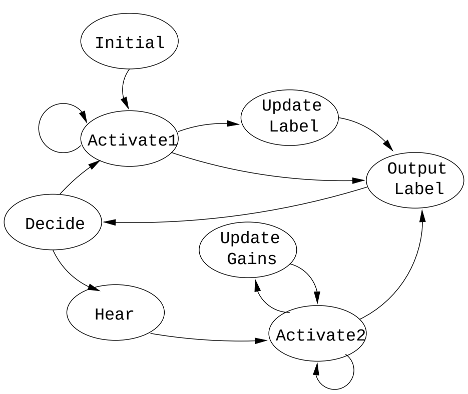


Figure 5: State machine of a `Concept` behavior.

`Activate1` computes the behavior activation. If the activation is above a threshold, it transitions to `OutputLabel`, whence the vocal label of the behavior is sent out. Otherwise, the next state remains the same, unless the condition for label update is satisfied. Then the transition function leads to the `UpdateLabel` state.

5.1.1 *Attribution of label*

When new speech input arrives, this activates the `Hear` state of every `Concept`, which attempts to match the heard phonemic string to the behavior's vocal label. The match value is computed based on the vocal label as a template against which to match, the confidence value for the behavior's own vocal label, and an empirically determined global phoneme confusion matrix:

$$BestMatch(in, template) = max(M(in, template)) \quad (1)$$

where $M(in, template)$ is a vector whose elements are best matches given a certain window size (window size is measured in phonemes and not in single characters):

$$M(in, template) = \frac{1}{n}\begin{bmatrix} M_m(in, template) \cdot m \\ M_{m+1}(in, template) \cdot (m+1) \\ \cdots \\ M_n(in, template) \cdot n \end{bmatrix} \quad (2)$$

where m is the minimum allowed window size and

$$n = min(MAX\_WINDOW\_SIZE, length(template))$$

The minimum window is set so that spurious one-phoneme matches are discarded. The matching measures returned by $M_i()$ are scaled by the window size in order to bias in favor of longer substrings. $M_i()$ itself uses the standard brute-force string searching algorithm.

`Hear` automatically transitions to `Activate2` which computes the activation level of the behavior based on the match determined earlier and the values of the behavior's receptors. If a discrepancy between the response to receptor values and the response to the speech input is above a threshold, and the confidence in the vocal label is high, `Concept` transitions to the `UpdateGains` state.

If the activation level reaches above a threshold, the behavior transitions to the `OutputLabel` state, which sets the behavior's priority to a value dependent on its activation and the confidence of the match, sets the output priority to the behavior's priority, and outputs an inhibitory signal at that priority level. The inhibitory outputs of each concept connect back to the `HeardThis` behavior. Such a signal received by `HeardThis` propagates further to `ConceptMap` and inhibits the default creation of a new `Concept` when a new string of phonemes is heard (see section 5.2).

5.1.2 Updating of label and confidence

Whether or not the vocal label of a `Concept` should be updated is determined by a combination of conditions on the `Concept`'s confidence C, the confidence C_{heard} in the accuracy of the heard phonemic string, the value of the best match between them and the current activation A of the `Concept`. The update label value U_L is compared against a threshold to decide whether an update should take place (where k is a scaling constant, chosen arbitrarily, and θ the activation threshold, d is the distance measure between the two strings in the result of the $BestMatch()$ call):

$$U_L = \begin{cases} k \, \frac{C_{heard}}{C} \, d & : \quad A > \theta \, , \, C \neq 0 \\ 0 & : \quad otherwise \end{cases} \tag{3}$$

If the `Concept` is active, then U_L will be proportional to the confidence attached to the heard string, and inversely proportionate to the goodness of the match and to the level of the `Concept`'s own confidence. If it is not active, there is no reason to update the label.

These computations depend on the confidence measure on the input speech string. This is computed inside `HeardThis` in the following way:

$$C_{heard} = default \; - \; \frac{1}{n} \sum_{n} \frac{1}{d_n + 1} \, C_n \tag{4}$$

where $default$ is a constant, d_n is a distance measure of the match between $heard$ and the n'th template, C_n is the confidence for the n'th template, and the sum is taken over all templates that had at least a minimal match to the heard string. The individual C_n and d_n are transmitted to the `HeardThis` behavior through incoming connections. N of these connections are created and we then let the active `Concept` behaviors compete to overwrite those connections only.

Note that C_n are used to compute C_{heard} but C_{heard} is needed to determine the update rule for the vocal label of concept n. However, this situation does not start an infinite regression in our implementation, as there is a clear time difference between the activation of the two connections, so that when C_{heard} is received back at `Concept`, the behavior has switched to another state as shown in figure 6.

5.1.3 Updating of parameters

When a behavior is created, the gains are set to default values, which must be updated in state `UpdateGains` to arrive at some consistent representation of releaser properties for that behavior. The signals that the `Concept` has access to at this point in the state machine include:

- its activation A
- the result of the matching process between its vocal label and the input phonemic string
- its confidence in its vocal label C
- a measure of confidence in the accuracy of the string heard C_{heard}
- the current presence or absence of relevant releasers

From this information the behavior derives a reinforcement signal U_G computed as follows, where d is the distance measure returned by $BestMatch()$, which is equal to 0 when the match is perfect:

$$U_G = \begin{cases} - \, k \, C_{heard} \, C \, d & : \quad A > \theta \\ \frac{C_{heard} \, C}{d \, + \, 1} & : \quad otherwise \end{cases} \tag{5}$$

This rule allows for both Type I and Type II errors: when the behavior is active even though something other than its label is heard, and when the behavior remains inactive and its label was heard. However, in the first case reinforcement should be weaker since it is quite often the case that the speech the robot hears is not describing any immediate features of the environment. Therefore we scale this type of signal by a smaller k. This does not eliminate the problem, but biases the computation in a simple way. U_G is then used in the update rule for each of the gains in the vector:

$$G^i_{t+1} \; = \; G^i_t \; + \; \alpha \; R^i_t \; G^i_t \; U_G \tag{6}$$

This will not change the value of G^i (the ith element of the gains vector G) if its corresponding `Receptor` R^i was not responding.

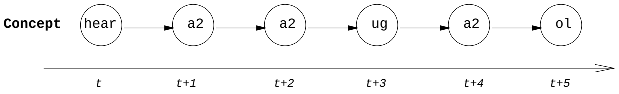


Figure 6: The timings of information transfer between `HeardThis` and any `Concept`. The state transitions ensure that confidence measures are only computed once.

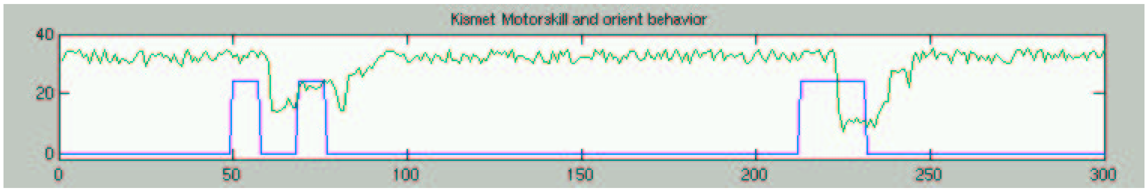


Figure 7: Activation of Kismet's motorskill internal variable (solid line) and the "orient" fixed concept (ragged line). Motorskill = 0 corresponds to the activation of the orienting behavior.

The simplicity of the update rules is possible because of the finite and very precisely described set of pre-existing releasers on Kismet. For a more scalable system, if we wish to maintain natural-time responsiveness, we will want to consider more sophisticated techniques.

5.2 Acquisition of novel concepts

`ConceptMap` is the single behavior which acts as a very high-level manager of concepts. Described as such it sounds like precisely the homunculus which we are trying to avoid by creating this distributed architecture of behaviors. However, the functionality of `ConceptMap` is very tightly tied in with that of `HeardThis` and should perhaps be implemented as two extra states within that behavior. That would perhaps be more acceptable in terms of the spirit of the project. The current implementation has singled out `ConceptMap` mainly for the sake of readability.

The vocal labels of concepts serve also as the key in the `ConceptMap` and may be used to remove a concept by finding the vocal label key. The map stores pointers to all `Concept` behaviors, i.e., all non-grunting, non-babbling protoverbal behaviors. The map is initialized to contain a small set of fixed concepts, which correspond to Kismet's existing behaviors.

Every time a new spoken input is heard, a new `Concept` will be generated by `ConceptMap`, with the heard phonemic string as its vocal label, unless inhibitory signals are received by `HeardThis` via any of its N input connections. When a new protoverbal behavior is automatically generated from `ConceptMap`, its vector of `Receptors` is initialized to include one for every kind of releaser that is present in the system. All receptor gains are set to a default value. This exhaustive assignment will be remedied later as the gains are updated individually within each behavior.

6. Preliminary Experiments

In (Varchavskaia, 2002), evaluation of `ConceptMap` workings, the acquisition and labeling of concepts was hindered by poor phoneme recognition and the often unpredictable nature of the robot's behaviors. Here we have attempted to remove some of these obstacles. We have created an initial small vocabulary of 20 words, chosen arbitrarily as most often used in interactions with the robot. This vocabulary was transcribed phonemically and used with ViaVoice exclusively for the purposes of more consistent phoneme recognition. This initial vocabulary was augmented with novel phonemic strings whenever novel concepts were added to the map. In this way, the robot is biased to hear the "words" it knows again, and we can eliminate much of the spurious creation of new concepts.

6.1 Labeling an existing behavior

During a 15 minute interaction, the human teacher attempted to teach Kismet labels for some of its original behaviors, such as orienting, search, and expressing an emotion. The words used for this purpose were not present in the initial ViaVoice vocabulary. Figure 7 shows the correspondence between the activation pattern of the "orient" concept and the robot's internal variable representing current motorskill. The table below shows the progression of label updates within the "orient" concept; the teacher wanted Kismet to associate the phrase "look here" with the orienting behavior.

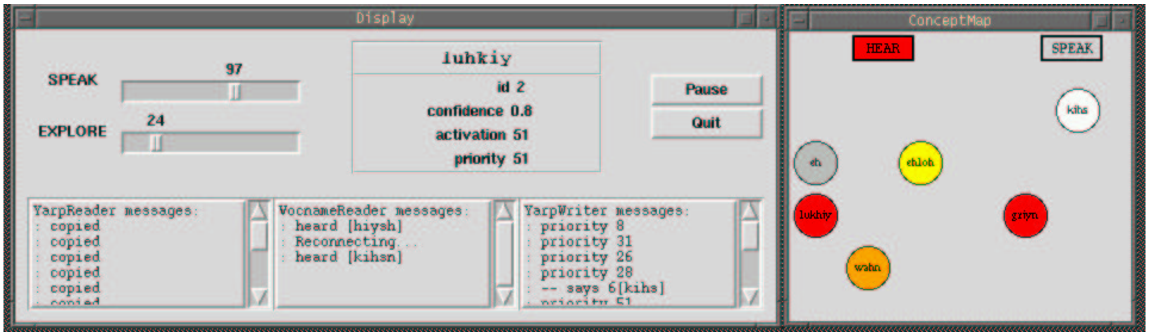
cycle:	last heard:	label:	C:	d:
200:	h uh k ih h	uh k iy	0.3	3.3
400:	h uh k iy	l uh k iy h	0.9	1.2
600:	l uw	h uh k ih	1.1	2.8

....

A cycle refers to one scan cycle of the vocal behaviors system, in which each behavior has exectuted for one state, and takes an average of about half a second. Figure 9 shows that appropriate labels were associated with fixed concepts, e.g., `/l uh k iy h/` with "orient", `/w ah n/` with "seek" (in this case, the robot was taught the label "wanna" for the seeking behavior).

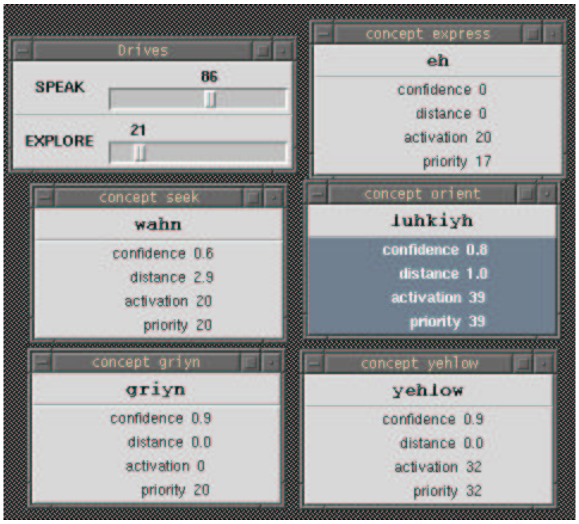
6.2 Acquiring a perceptual category

During another 15 minute interaction, the human attempted to teach Kismet the English words for the colors green and yellow through the following scenario. The teacher shows a toy of the corresponding color and says the word when Kismet's attention is



```
SPEAK 97
EXPLORE 24

luhkiy
id 2
confidence 0.8
activation 51
priority 51

YarpReader messages:
: copied
: copied
: copied
: copied
: copied
: copied

VocnameReader messages:
: heard [hiysh]
: Reconnecting...
: heard [kihsn]

YarpWriter messages:
: priority 8
: priority 31
: priority 26
: priority 28
: -- says 6[kihs]
: priority 51
```

Figure 8: Partial screenshot of an interaction. ConceptMap on the right shows current protoverbal behaviors colored by their priority value. Details of winning behavior also in the main window.



```
Drives
SPEAK 86
EXPLORE 21

concept express
eh
confidence 0
distance 0
activation 20
priority 17

concept seek
wahn
confidence 0.6
distance 2.9
activation 20
priority 20

concept orient
luhkiyh
confidence 0.8
distance 1.0
activation 39
priority 39

concept griyn
griyn
confidence 0.9
distance 0.0
activation 0
priority 20

concept yehlow
yehlow
confidence 0.9
distance 0.0
activation 32
priority 32
```

Figure 9: Partial screenshot of an interaction. Highlighted concept won the priority competition. "orient", "seek" and "express" are fixed concepts mirroring the activaiton of the robot's nonverbal behaviors. "yehlow" and "griyn" are new concepts created by the system.

focused on it. Then the toy is removed and only shown again if the robot shows a seeking behavior and vocalizes a "request" which sounds similar to the color label. The words used were transcribed as strings of phonemes for ViaVoice. Figures 9 and 8 show that the concepts with corresponding labels were created. However, we do not at present have a satisfactory evaluation method that would demonstrate that the labels correspond precisely to that perceptual category.

7. Discussion

The scope of the project was to provide a framework and a pragmatic, behavior-based approach to the problem of early concept and vocal label acquisition. Therefore, emphasis was placed on the architecture of the system rather than on the learning algorithms involved. Consequently, the methods used are often meant as placeholders for more sophisticated models.

Nevertheless, the preliminary results show that this approach may be fruitful. We have created a system of protoverbal behaviors, which operate for a purpose, namely the satisfaction of the robot's drives, including a drive for communication. The framework allows the "words" in the robot's protolanguage to have grounded function and meaning.

Currently we are focusing on developing appropriate evaluation methods to test the system. Future research will also involve integration of better phoneme recognition, through incorporation of the out-of-vocabulary model reported in (Varchavskaia et al., 2001), and a mechanism for word segmentation from spoken utterances. We should also conduct more experiments, including those with naive subjects to test claims of naturalness and meaningfulness of interaction.

Acknowledgements

Funds for this project were provided by DARPA as part of the "Natural Tasking of Robots Based on Human Interaction Cues" project under contract number DABT 63-00-C-10102.

References

Bloom, P. (2000). *How Children Learn the Meaning of Words.* Cambridge: MIT Press.

Breazeal, C. (2000). *Sociable Machines: Expressive Social Exchange Between Humans and Robots.* PhD thesis, MIT.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation RA-2.*

Fitzpatrick, P. (1997). A novel behaviour-based robot architecture and its application to building an autonomous robot sentry. Master's thesis, University of Limerick, Ireland.

Halliday, M. (1975). *Learning How To Mean: Explorations in the Development of Language.* Cambridge: MIT Press.

Harnad, S. (1990). The symbol grounding problem. *Physica D*, 42:335–346.

Nicolescu, M. N. and Matarić, M. J. (2001). Learning and interacting in human-robot domains. In White, C. and Dautenhahn, K., (Eds.), *Special Issue of IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, volume 31.

Oates, T., Eyler-Walker, Z., and Cohen, P. (2000). Toward natural language interfaces for robotic agents: Grounding linguistic meaning in sensors. In *Proceedings of the 4th International Conference on Autonomous Agents*, pages 227–228.

Pinker, S. (1999). *Words and Rules: The Ingredients of Language.* Basic Books.

Roy, D. K. (1999). *Learning Words from Sights and Sounds: A Computational Model.* PhD thesis, MIT.

Steels, L. (1999). *The Talking Heads Experiment, vol.I Words and Meanings.* Special pre-edition for LABORATORIUM, Antwerp.

Steels, L. and Kaplan, F. (2001). Bootstrapping grounded word semantics. In Briscoe, T., (Ed.), *Linguistic evolution through language acquisition: Formal and Computational Models.* Cambridge University Press.

Varchavskaia, P. (2002). Early pragmatic language development for an infant robot. Master's thesis, MIT.

Varchavskaia, P., Fitzpatrick, P., and Breazeal, C. (2001). Characterizing and processing robot-directed speech. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots.*

Wray, A. (2000). A protolanguage with no declaratives and no names. In *Proceedings of the 3rd Conference on the Evolution of Language.*

Yanco, H. (1994). Robot communications: issues and implementations. Master's thesis, MIT.