Adaptive Behavior

http://adb.sagepub.com

From Deliberative to Routine Behaviors: A Cognitively Inspired Action-Selection Mechanism for Routine Behavior Capture

Sonia Chernova and Ronald C. Arkin

Adaptive Behavior 2007; 15; 199

DOI: 10.1177/1059712306076255

Adaptive Behavior

From Deliberative to Routine Behaviors: A Cognitively Inspired Action-Selection Mechanism for Routine Behavior Capture

Sonia Chernova[1], Ronald C. Arkin[2]
[1]Computer Science Department, Carnegie Mellon University
[2]College of Computing, Georgia Institute of Technology

Long-term human–robot interaction, especially in the case of humanoid robots, requires an adaptable and varied behavior base. In this paper, we present a method for capturing, or learning, sequential tasks by transferring serial behavior execution from deliberative to routine control. The incorporation of this approach leads to the natural development of complex and varied behaviors, with lower demands for planning, coordination and resources. We demonstrate how this process can be performed autonomously as part of the normal function of the robot, without the need for an explicit learning stage or user guidance. The complete implementation of this algorithm on the Sony QRIO humanoid robot is described.

Keywords action selection · developmental robotics · routine behavior · humanoid robot · robot architecture

1 Introduction

It is well known that there are high expectations of anthropomorphic robots with regard to natural behavior in their interactions with humans. Furthermore, interaction over extended periods of time requires that the robot acquire new skills, adapt to its surroundings, and change its behavior in response to different situations in order to appear natural and interesting. In this article, we address the issue of compelling long-term interaction by enabling the robot to adapt through the capture and execution of routine behaviors.

Routine behavior is defined as the habitual performance of an established procedure or task. Routine behavior occurs in a reactive manner, sometimes without awareness, and requires far less focused attention than conscious and highly supervised task execution, or deliberative behavior.

In humans, the classification of a task into one of these two categories is dependent on the familiarity of the task, as well as other psychological aspects such as perceived dangers or complex decision-making (Norman & Shallice, 1986). Over time, most behaviors are typically transferred from deliberative to routine execution. This type of adaptation is believed to be fundamental to our development because of the limited nature of human attention.

In this article, we demonstrate a similar adaptational mechanism for robotic systems. We present a method for capturing, or learning, robotic tasks and

Correspondence to: Sonia Chernova, Computer Science Department, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA. *Tel.*: +1 412 268 2601
E-mail: soniac@cs.cmu.edu

(2007), Vol 15(2): 199–216.
DOI: 10.1177/1059712306076255

Figure 1 The Sony entertainment robot QRIO.

transferring their execution from deliberative to routine control. Specifically, we focus on sequential tasks, those that involve series of consecutive actions. We present a method based on the repeated execution of a task, which allows the robotic system to learn habitual execution, so as to require less planning, coordination and resources. We demonstrate how this process can be performed autonomously as part of the normal function of the robot, without the need for a specific learning stage or user guidance. The incorporation of this method results in the more varied and adaptable behavior that is so important for successful long-term interaction.

This research was conducted on a fully autonomous robotic humanoid system, the Sony QRIO entertainment robot (Figure 1), designed for long-term human–robot interaction. All experiments were performed on the physical platform without simulation.

1.1 Psychological and Neuroscientific Motivations

Action selection, the problem of selecting what to do next, has been studied extensively in humans in the areas of psychology, physiology, neurology and other related fields (Cleeremans, 1993; Keele, Ivry, Mayr, Hazeltine, & Heuer, 2004; Sun & Giles, 2001). Our approach is motivated by the contention scheduling (CS) model for action selection, originally proposed by Norman and Shallice (1986); see also Cooper, Shallice, and Farringdon (1995) and Cooper and Shallice (1997, 2000). CS combines elements of automatic and deliberative action execution, encompassing both action selection for routine actions as well as deliberative planning and execution.

The original inspiration for CS, and other serial behavior models, can be traced back to early work by Lashley (1951). Lashley stated, consistent with the CS model, that a parallel set of chunk actions is activated even before action is produced. More recently, Houghton and Hartley (1995) revisited Lashley's early work on parallel models of serial behavior, and employed the concept of schemas as the basis for producing sequential behavior, providing new evidence to support his claim. This inspired the concept of activation levels (ALs) that is currently used in one form for its action-selection mechanism in the QRIO architecture.

In other work, Cisek (2001) contends that animals have two pragmatic concerns: action specification and action selection. Specification allows multiple processes to be primed for action, which are based on spatial information of the agent and its relationship to world objects, while selection reduces this set to a unique behavior for enactment based upon the nature and identity of the environmental objects. To the best of our knowledge, however, this model has yet to be exported to a robotic system in its native form.

One recent neuroscientific study (Badgalyan, 2000) gives strong support to the Norman–Shallice model of central deliberative control by identifying regions in the brain where such activity occurs: specifically, the cingulate and pre-frontal cortical regions. The statement that the deliberative system (DS) is recruited only when the action is conscious helps us to understand the non-dominant relationship between the supervisory attentional system (SAS) and the contention scheduler, and helps to ground us in designing a suitable interface between planning and control for QRIO.

Other neuroscientific studies implicate the basal ganglia in action selection as well as action gating via disinhibition (Prescott, Gurney, & Redgrave, 2003). Their notion is that sensory requests arrive at the basal ganglia in the form of requests for access to the motor control system, which utilizes multiple selection mechanisms to service those requests. A model of this system was embedded in a mobile robot and tested, but more for the ability to evaluate it as an explanatory model of the human action-selection mechanism as opposed to an efficient robot controller.

Several implementations of the CS model have also been developed, including one by Cooper and Shallice themselves (Cooper & Shallice, 2000). However, most work so far has focused on the independent implementation and study of the two behavior mechanisms for the control of routine and non-routine behaviors. The problem of transference of behaviors and skills from deliberative to routine control is much less studied as it requires a functional model of both systems.

We are aware of only two projects involved in the study of routine behavior capture for robotic systems relating to the CS model. One is the work of Garforth, Meehan, and McHale (2000), where the CS model is implemented as a large-scale neural network. Garforth et al. demonstrate the transfer of skills from deliberative to routine control using a simulated robot under the task of avoiding distracting stimuli in order to fully complete a primary task. This study differs significantly from the work presented here, not only in the implementation of the system, but also in its functionality and target behavior. Specifically, the related work is concerned with only single behaviors instead of sequential tasks, and does not include an emotional model.

The second related work, published by Cooper and Glasspool (2001), focuses on acquiring environment–action associations through unguided exploration in the environment and reinforcement learning. This work differs significantly in that learning is performed only at the lowest levels, pairing environmental features with basic actions. It is independent of any DS and the algorithm does not represent any higher-order behavior sequences.

Our work is complementary to the Cooper and Glasspool approach, consisting of a mechanism for capturing higher-order behavior sequences that comprise more complex tasks. Over time, sequential tasks executed through deliberation can become routine, such that the mechanism of their execution is shifted to the reactive behavior level and deliberative planning is no longer involved in the execution of the sequence. This separation of routine and non-routine behaviors allows different methods to be applied to each, while reducing the load on the planner and freeing system resources. Our approach is the first complete implementation of the routine behavior capture mechanism at this level, and we demonstrate its effectiveness using the fully functional behavior model of the QRIO robot.

In the rest of the article we present an overview of QRIO's behavior system, the emotionally grounded (EGO) architecture, followed by a detailed description of the behavior selection algorithm. We then present the routine behavior serialization method, which allows the transference of routine behavior control from the deliberative to the reactive layer of the behavior system. Finally, we describe the results of a series of experiments.

2 Behavior System Overview

The autonomous behavior control architecture of the QRIO robot, called the EGO architecture, is designed for long-term human–robot interaction based on an ethological behavior model (Arkin, Fujita, Takagi, & Hasegawa, 2003). Detailed descriptions of its various components appear in a series of previous publications (Fujita, Kurolki, Ishida, & Doi, 2003; Hoshino, Takagi, Profio, & Fujita, 2004; Sawada, Takagi, & Fujita, 2004a; Sawada, Takagi, Hoshino, & Fujita, 2004b; Tanaka, Noda, Sawada, & Fujita, 2004), and in this section we provide only a brief description of the individual software components related to this work. Figure 2 presents an overview of the system.

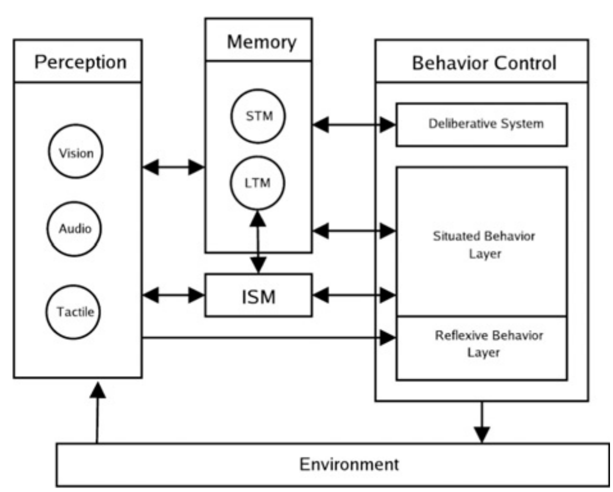

Figure 2 Overview of the EGO architecture components.

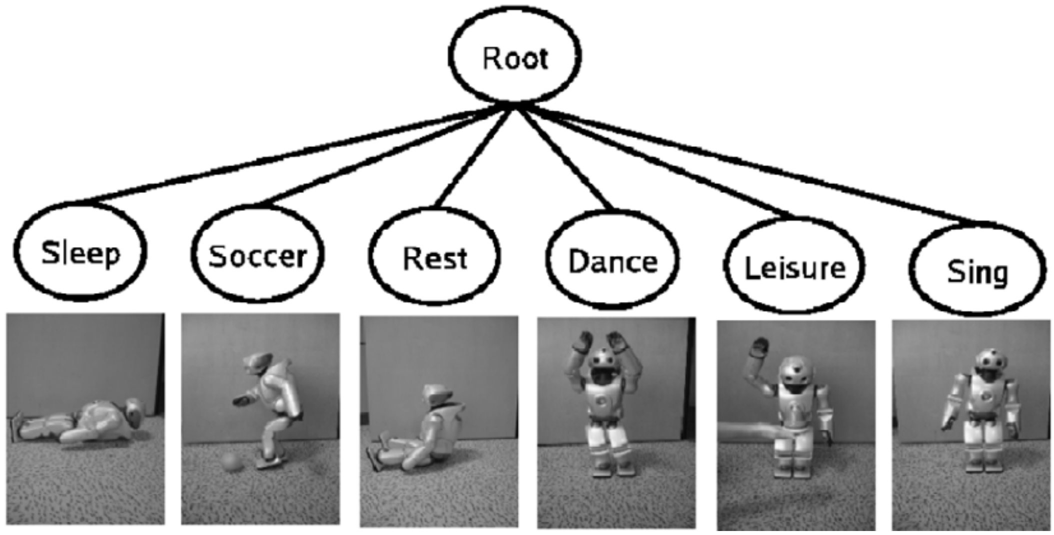

Figure 3 An example of the SBL behavior tree.

2.1 The EGO Architecture

The perception component of the system processes data on three different sensor channels: visual, auditory and tactile. Information about the environment perceived through the robot's sensors is integrated into the short-term memory (STM) segment of the memory component, computing information such as object location and sound source, while assigning IDs to these perceptual events. Using the robot's kinematic data, the STM also tracks and maintains the position of objects located outside the current field of view of the robot.

The long-term memory (LTM) component associates perceived information with the internal state of the robot. This allows the robot to identify people through face and voice recognition, as well as to recall previously made associations and emotions felt about the person.

Variables related to the internal state of the robot are maintained by the internal state model (ISM). Example state variables include *nourishment*, *sleep*, *fatigue* and *vitality*. Changes in the robot's internal state occur with the passage of time, and as a result of external stimuli or the robot's actions. While some variables can be grounded on a physical sensor, such as battery charge level, others represent more abstract values.

Three different behavior control modules are responsible for behavior selection. The reflexive behavior layer (RBL) regulates behaviors that require a quick response time, such as being startled. The situated behavior layer (SBL) controls reactive and homeostatic behaviors. The DS performs deliberative planning and control of sequential behaviors. Within the SBL, which is the focal point for this research, behaviors are organized in a tree-structured network of schemas (Figure 3). Action selection is conducted when schemas in the network compete for activation based on their relevance and resource requirements. The specific details of this behavior selection mechanism are described in the following section.

2.2 Behavior Selection

The behavior cycle is executed at a rate of 2 Hz. During each cycle, every behavior calculates a fitness value called the AL, which indicates the relevance of that behavior in the current situation. The AL is calculated based on the external stimuli and internal state of the robot, as well as intentional values provided by the DS. Details of the DS are discussed in Section 4.1.

Behavior selection occurs using a greedy policy: the behavior with the highest AL is selected first. Other behaviors, from highest to lowest AL value, can then be selected for concurrent execution as long as their resource demands do not conflict with those already chosen. A behavior's resource demands are the physical robot parts (torso, head, etc.) and virtual resources (speech, etc.) needed for the execution of the behavior. A more detailed discussion concerning parallel activation and the propagation of AL values through the behavior tree can be found in Hoshino et al. (2004).

In the current implementation, AL values are strictly positive and can have arbitrarily high values. Alternative approaches could include normalizing or bounding all values to some range. Because the values are compared on a relative scale instead of an absolute scale, all of these approaches will result in the same outcome.

The AL function, used to calculate the AL value, plays a crucial role in determining which behaviors are selected. In effect, different values in the AL function control the overt personality and behavioral manifestation of the robot, making it of critical importance in the development of an entertainment robot system. In the following section we describe the full details of this implementation.

3 Activation Level Evaluation

Our new formulation of the AL function builds upon the previously existing AL function of the EGO architecture (Hoshino et al., 2004; Sawada et al., 2004b), while adding fundamental components inspired by the CS model (Cooper & Shallice, 2000; Cooper et al., 1995). The pre-existing AL function was based upon an ethologically inspired model (Arkin et al., 2003) and contained elements balancing internal motivations and external stimuli.

We have extended this function to now include a resting level and random noise, both of which are key features governing the CS model (Cooper & Shallice, 2000; Cooper et al., 1995). Most importantly, we have added a new mechanism, which we call self-excitation (SE). Although inspired by the concepts of self-activation and lateral inhibition present in the CS model, in a way this replaces both while providing new abilities and flexibility to the system.

The new AL value is calculated as a weighted sum of the following five components.

Previously existing components:

- motivation value (*Mot*);
- releasing value (*Rel*).

Contributed components:

- resting level (*RL*);
- random noise (*Noise*);
- self-excitation value (*SE*).

Each of these contributing factors are described below.

3.1 Motivation Value

The motivation value is derived from the internal state of the robot, and embodies its instinctual drive. It is calculated as a weighted sum of individual instincts, *Ins*[*i*]. Each instinct value corresponds to an internal state *i*, and expresses the robot's current desire with respect to that state (Hoshino et al., 2004; Sawada et al., 2004a).

Example values for the *nourishment* internal state are shown in Figure 4. The instinct associated with *nourishment* is high when the internal value of *nourishment* is low, signifying that the robot has a desire to satisfy this need and raise the *nourishment* level. The less *nourishment* there is, the greater the desire to obtain it. In the case when the *nourishment* level is high, the instinct value is low to signify satisfaction. It is also possible to have a negative instinct value, signifying that the robot has exceeded the desired level for that internal state. In our example, this phenomenon symbolizes overeating; if the *nourishment* level is too great, the instinct or desire to eat becomes negative.

The function relating each instinct to its corresponding internal state is designed on an individual basis, as described in Sawada et al. (2004a). The motivation value for each behavior is calculated as a weighted sum of the robot's current instincts

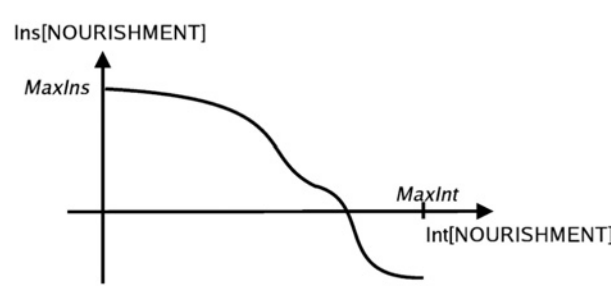

Figure 4 Curve expressing the relationship between intention and instinct for the internal state *nourishment*.

$$Mot_{\mathrm{Beh}} = \sum_i WM_{\mathrm{Beh}}[i] \cdot Ins[i], \tag{1}$$

where $WM_{\mathrm{Beh}}[i]$ is the motivation weight associated with instinct $Ins[i]$ for that behavior. The weights are used to control which instincts motivate each behavior. For example, an instinct based on *nourishment* would motivate an eating behavior, but may not affect the desire for sleep or exercise.

3.2 Releasing Value

The releasing value denotes the expected satisfaction associated with an internal state that the robot would achieve from executing a behavior. It is composed of the current satisfaction value, $Sat[i]$, which is based on the internal state of the robot at this time, and an expected satisfaction, $ESat[i]$, which is calculated based on the internal state and the properties of external stimuli (Sawada et al., 2004b).

Expected satisfaction is calculated by predicting the change in internal state as a result of performing a behavior. Consider the *nourishment* example, as shown in Figure 5. From the figure, we see that if the *nourishment* level is low, raising it leads to increased satisfaction. However, if the *nourishment* level is currently high, eating further can lead to overeating and a decrease in the satisfaction value. The lowest levels of satisfaction result from extreme hunger and extreme satiety.

Satisfaction values can also depend on the characteristics of external stimuli, such as object type, size or distance. For example, it is possible to expect more satisfaction from interacting with a friend than a stranger. In the current implementation of our system, satisfaction functions are designed on an individual basis for each behavior.

The releasing value is calculated based on the satisfaction values using

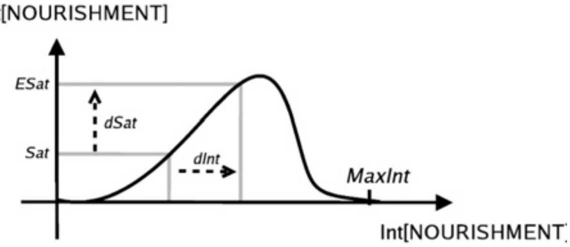

Figure 5 Curve expressing the relationship between intention and satisfaction for the internal state *nourishment*.

$$dSat[i] = ESat[i] - Sat[i] \tag{2}$$

$$Rel_{\mathrm{Beh}} = \sum_i WR_{\mathrm{Beh}}[i] \times (W_{\mathrm{dSat}} dSat[i] + (1 - W_{\mathrm{dSat}} ESat[i])). \tag{3}$$

Here, $dSat[i]$ denotes the expected change in satisfaction, $WR_{\mathrm{Beh}}[i]$ is the releasing weight associated with internal state i, and W_{dSat} is the weight of the expected change in satisfaction against the final expected satisfaction value. For further details on motivation and releasing values, and their involvement in behavior selection, see Sawada et al. (2004a).

3.3 Rest Level

The rest level value establishes a baseline activation for behaviors having no additional input. In the absence of any internal or external influences, the AL value tends toward the rest level. Its function is to permit low-level differentiation of behaviors based on priority due to the assignment of different resting levels.

In the case when a behavior is active and there are no internal or external stimuli, the drop of the AL to rest level is controlled by a decay function called the persistence function (Figure 6). This feature mimics the mechanism by the same name proposed by Norman and Shallice (1986). Its purpose is to promote the continuity of the AL of active behaviors.

In the case that a behavior is already active, it is natural that the desire to perform the activity should persist for some time after all stimuli disappear. For example, if the robot is playing soccer and loses sight of the ball, the desire to play soccer should persist for some time, allowing the robot to search for the ball. However, the probability that another behavior is selected should increase over time.

Persistence is implemented through a decay function which decreases the status SE value over time. The decay continues until either the rest level value is reached or another behavior is activated.

3.4 Noise

The random noise parameter adds normally distributed random noise to the AL in order to break ties between equally competing behaviors and add variability to the behavioral display. The magnitude of the noise is a controllable parameter, usually proportional

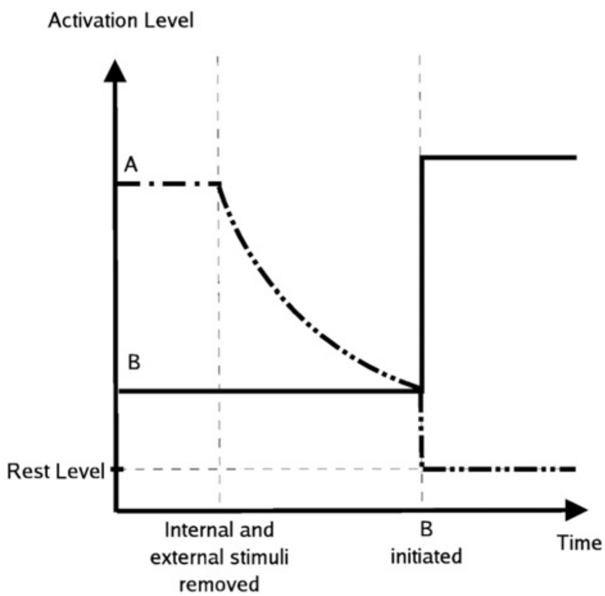

Figure 6 An idealized graph of the ALs of two behaviors, demonstrating the effect of the persistence function. In the absence of any internal or external stimuli, the AL of the active behavior decays until another behavior is activated as a result of a higher AL value. Activation of behavior A drops to the rest level.

to the strength of the AL value. Figure 7 shows noisy ALs of three competing behaviors. The noise component is omitted in other example figures in this section in order to demonstrate the underlying concepts more clearly.

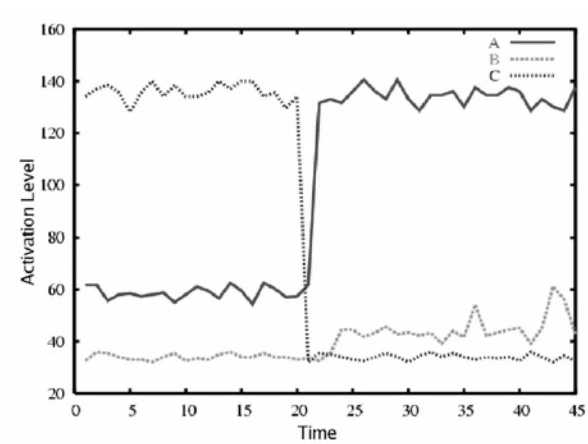

Figure 7 Graph of actual AL values including the noise parameter.

3.5 Self-Excitation

The SE concept, and the underlying function used for setting this value, comprise one of the main contributions of this work. SE is composed of two factors, status self-excitation (SSE) and routine self-excitation (RSE), which play two different and important roles in AL regulation.

$$SE = SSE + RSE. \tag{4}$$

SSE serves as a mechanism for minimizing the risk of rapid oscillation between behaviors. Its value is dependent upon the current operational status of the behavior, which represents whether the behavior is currently active, not active, or transitioning in or out of the active state. Different levels of excitement are associated with each status.

A behavior that is not active experiences very little or no excitement. When the behavior becomes active, its excitement level rises until the completion of the behavior, increasing the overall AL value. This ensures that the AL of an active behavior has an additional margin of separation from the other behaviors. The AL of other behaviors must overcome this margin in order to become active. Figure 8 shows the ALs of three competing behaviors and demonstrates the effect of SSE.

This mechanism is designed to increase the likelihood that each behavior runs to completion without

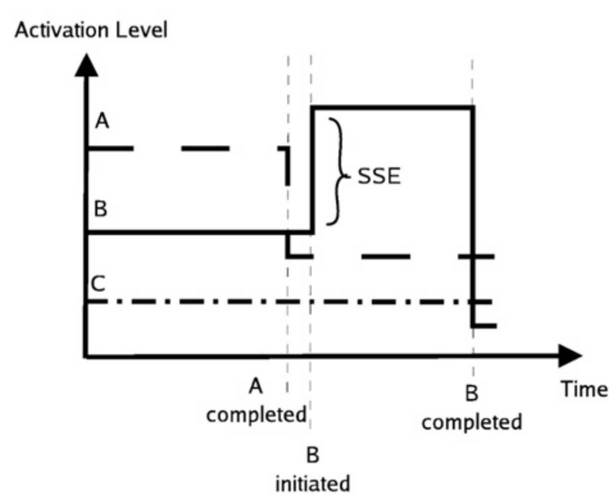

Figure 8 An idealized graph of the ALs of three competing behaviors, demonstrating the role of SSE in AL regulation.

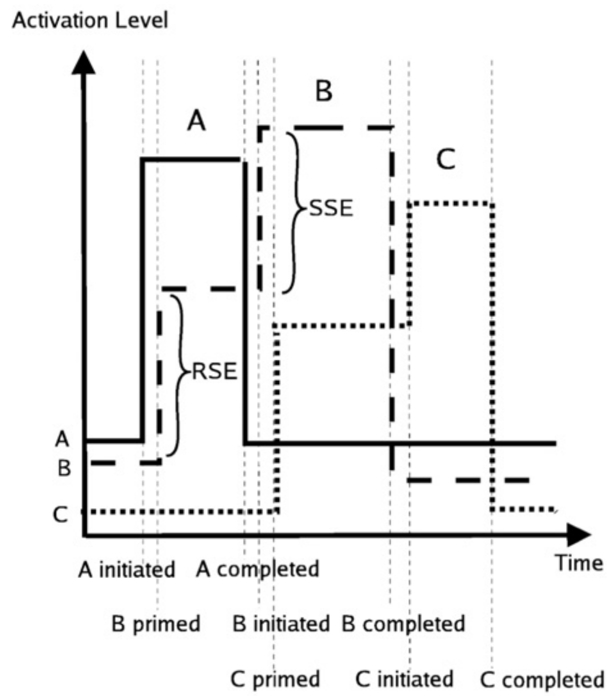

Figure 9 An idealized graph showing the AL of a captured routine sequence consisting of three behaviors executed via RSE.

interruption. This is a desirable characteristic as it prevents behavioral dithering, the phenomenon of behaviors thrashing back and forth between themselves as they compete for execution. Note that the currently active behavior can still be interrupted if the AL of a competitor surpasses the excitation margin.

RSE controls the activation of routine behavior sequences. Its behavior is based on sequences of behaviors that were captured as a result of repeated execution of some task.

Figure 9 demonstrates the effect of RSE on a captured routine sequence of behaviors $A \rightarrow B \rightarrow C$. In the captured sequence, each behavior knows its predecessor in the chain. RSE works by increasing the AL of each behavior when its predecessor becomes active, a process called priming. In our example we see that the AL value for *B* rises as its predecessor *A* becomes active. A similar relationship exists between *C* and *B*.

Intuitively, this can be interpreted as a behavior anticipating or predicting being next in the sequence. The resulting higher AL increases the likelihood, although does not guarantee, that all the behaviors in the sequence will be executed in the captured order. Note that the RSE value is usually set so as not to overcome the SSE margin, otherwise the primed behavior will activate early, interrupting the execution of its predecessor. Overall, the excitation components work together to achieve continuous and complete behavior sequence execution. Further details of the RSE implementation are discussed in Section 4.

3.6 Activation Level Computation Summary

The complete AL function is summarized in Equations 5 and 6. The motivational and releasing components are combined in *MR*, where W_M sets the weight or importance between the two. W_{SE} provides a similar balance between SE and *MR*.

$$MR = W_M Mot + (1 - W_M) Rel \tag{5}$$

$$AL = W_{SE} SE + (1 - W_{SE}) MR + RL + Noise. \tag{6}$$

The two weight parameters of the AL function, W_M and W_{SE}, control the emergent behavioral pattern and apparent personality of the robot. A high W_M results in the robot exhibiting self-centered behavior aimed at satisfying its own internal state. This type of behavior would likely be perceived as selfish or unfriendly by humans as the robot may ignore their presence or attempts at interaction. A low W_M value would result in the opposite effect where the robot would be overly responsive to external stimuli and interaction. W_{SE} controls how likely the robot is to complete a task it has begun. It affects both short independent behaviors and the completion of longer chains of sequential tasks. Algorithms for dynamically adjusting these values still need to be explored; the current implementation relies on fixed values.

4 Routine Behavior Capture and Execution

Routine behavior is defined here as the repeated execution of the same task or sequence of tasks over a prolonged period of time. Over its lifetime, QRIO will complete many repeated tasks. In this research, we are specifically interested in higher-order routine tasks composed of a number of sub-behaviors executed in a

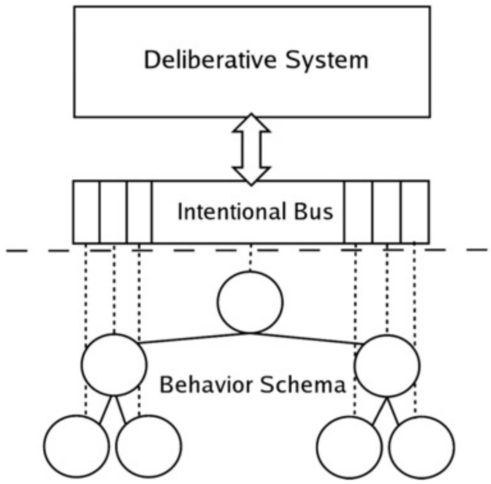

Figure 10 Overview of the interaction between the DS, the intentional bus and the SBL behavior schema tree.

specified order. For example, the task of playing soccer requires the robot to find the ball, approach and kick it. These sub-behaviors must be executed in a specific sequence in order for the overall task to be successfully completed.

In this section, we first describe the mechanism by which normal sequential behavior is executed through deliberative intention. We then present a novel approach for capturing, or learning, these repeated behavioral sequences and executing them subsequently as routine behaviors.

4.1 Deliberative System

In this architecture, deliberative control of the robot is performed by the DS via the intentional bus (Figure 10; Ulam & Arkin, 2006). The intentional bus forms a gateway between the reactive and deliberative layers in the architecture. Its purpose is to convert the high-level, goal-oriented tasks generated by the DS into intentional bias that serves to influence behavioral activation in the reactive layer. These biases can be combined with the existing AL computations (Equation 7) in order to guide the robot towards accomplishing the goals of the DS:

$$AL_{\text{total}} = AL + \text{intentional bias}. \tag{7}$$

The intentional bus provides three major functions for the DS:

- it serves as a repository of information about the underlying behavior level, maintaining information about the current status of each behavior as well as their AL;
- it biases the ALs of designated reactive behaviors in response to requests emanating from the DS;
- it maintains appropriate levels of intentional bias despite changes in ALs of the behaviors.

When the DS generates and then executes a plan for the robot, the intentional bus responds by sending intention, or bias, to the first behavior in the planned sequence. The strength of the bias is encoded in the plan representation itself, and conveys the importance of that behavior in the sequence. A weak intentional bias implies that the behavior has a low priority, which may result in its execution being interrupted, deferred, or perhaps even skipped should higher priority activities be in play.

While the behavior is performed by the robot, the intentional bus monitors its progress and maintains appropriate intentional bias levels. When the behavior completes, the focus is shifted to the next behavior in the sequence and the process is repeated. In this way, the intentional bus activates each of the behaviors in the plan in the specified order. Figure 11 shows the ALs of a sequence of four behaviors executed by the DS.

4.2 Routine Behavior

While the execution of deliberative plans is controlled by a centralized, higher-level system, the capture and execution of routine behaviors occur entirely at the reactive layer. Information between schemas is shared via a structure called LogMemory, which records the status and intention value of each schema. The capture of routine sequences occurs independently for each behavior; each schema only records and learns behavior sequences in which it plays a part. Figure 12 provides an overview of the serialization process.

Using information from LogMemory, each behavior creates an intentioned schema history (ISH), which maintains information about any currently active deliberative plans. Statistics maintained by the ISH include the order of the behaviors in the sequence and the value of the intentional signal sent via the intentional bus. If no plans are being executed by the DS, the history remains empty.

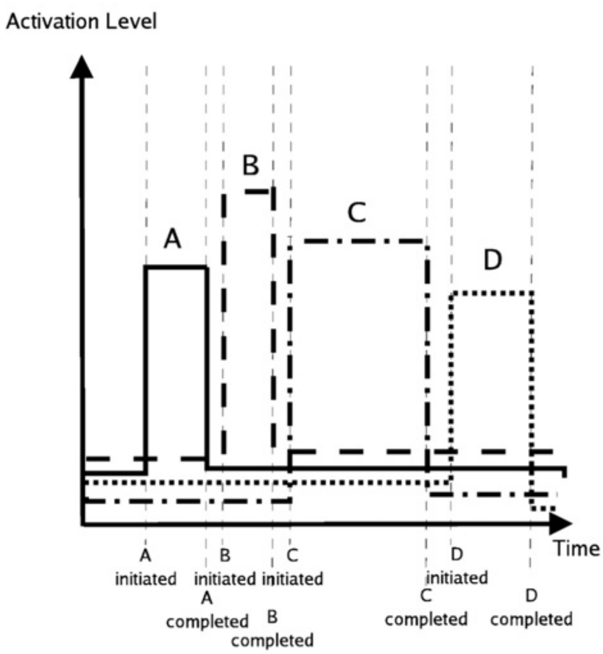

Figure 11 An idealized graph showing the ALs of a sequence consisting of four behaviors executed under the control of the DS.

When a specific behavior is deliberately activated by the intentional bus, it queries the ISH to determine the behavioral sequence prior to its activation. When the active behavior is the first in a planned sequence, the ISH is empty. If the behavior is not the first of a planned sequence, then the ISH contains a summary of the entire plan executed up to this point. The behavior listed as the most recent in the ISH, the previous step in the plan, is called the trigger schema, and this occurrence is called a triggered sequence.

When a triggered sequence occurs, the trigger schema is compared with a list of previously encountered trigger schemas. If this trigger schema has never been previously encountered, the schema is entered into the candidate captured routine list (CCRL). This list maintains a record of behaviors that have preceded the currently active behavior in previously executed plans. The schemas listed here have no effect on the AL, but instead serve as a local memory bank of previous experiences for the behavior.

When a triggered sequence occurs that involves a previously encountered trigger schema, information relating to that schema is updated. Statistics maintained about each schema include the average intentional bias value used to trigger the schema and the total number of times the schema has been active.

As stated above, trigger schemas on the candidate list do not yet have an effect on the RSE and AL, and do not comprise captured behaviors. A trigger schema must pass certain requirements before being captured and considered a part of a routine activity. Once captured, the schema is upgraded from the CCRL to the captured routine list (CRL).

Each behavior maintains its own CRL and monitors the status of the associated trigger schemas through LogMemory. When a trigger schema becomes active, the behavior primes itself for activation in anticipation of being next to execute, as seen in Figure 9. When priming occurs, the RSE value increases, raising the overall AL of the behavior. The amount by which the RSE increases is proportional to the average intentional bias (IB_{avg}) received from the intentional bus.

The process by which a trigger schema, symbolizing part of a sequence, achieves routine behavior status is important for the success of the system. Capturing

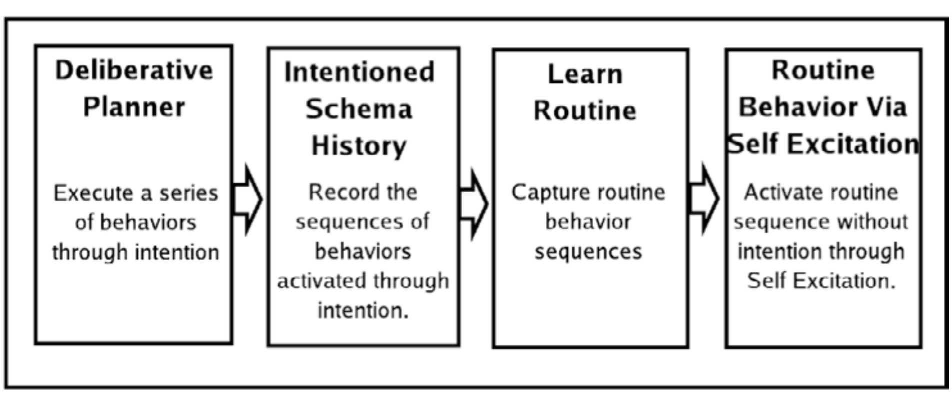

Figure 12 Overview of the behavior serialization process.

sequences too quickly may result in undesired behavior by the robot, while capturing too slowly will make the whole process ineffective, as relatively few routine behaviors will be learned. We have tested three different approaches to this problem, although other methods clearly exist.

1. **Simple thresholding**. The sequence pairing must occur some minimum number of times before it is considered routine.
2. **Pair frequency thresholding**. In addition to the simple thresholding criteria, the behavior must follow the trigger schema in a significant proportion of seen plans. More specifically, the ratio of the number of times the behavior is activated following the trigger schema relative to the total number of times the trigger schema is active must pass some threshold. If the pairing occurs only sporadically, and the majority of the time the trigger schema is followed by some other behavior, then this pairing does not have a strong routine bond and is not captured.
3. **Convergence thresholding**. In addition to the simple thresholding criteria, the recorded average intention value must stabilize or converge. This ensures that the RSE, which is calculated based on the average intention value, accurately simulates the intentional bias signal.

The RSE value replaces the role of the intentional bias in its control over generating the behavior sequence after the routine has been captured. It is therefore natural that the RSE is calculated based on the average intentional bias value. The intentional bias average (IB_{avg}) is maintained for each behavior pair individually, because the strength of the bias signal conveys the strength or importance of the sequence pairing. The following methods have been tested for calculating the RSE value.

1. **Static RSE**. The RSE is equal to IB_{avg}.
2. **Likelihood RSE**. The RSE is equal to IB_{avg} scaled by the likelihood of the routine pairing occurring. This results in a higher SE value for more likely pairings, increasing the probability of their occurrence.
3. **Stochastic RSE**. The RSE value is calculated using a stochastic method based on the likelihood of the pairing occurring. Note that, in the current implementation, the individual behavior schemas share only a limited amount of information; specifically, the ALs and RSE values are not shared. Therefore, the RSE value is probabilistically chosen between the full IB_{avg} value and the value of IB_{avg} scaled by the likelihood of the routine pairing occurring. Other stochastic approaches could also be applied.

The final issue that must be addressed is at what point a behavior sequence becomes truly routine. Despite learning the appropriate sequencing, anticipating activation and setting RSE, this goal is not achieved until deliberative planning is no longer directly involved in the execution of the sequence. To accomplish this, the DS must be notified when a routine is captured.

Once notified, the DS can choose to stop controlling the plan through intention and shift to an attentional method that merely triggers the sequence rather than constantly overseeing it (Ulam & Arkin, 2006). This process can be compared to a parent who, when teaching their child to ride a bicycle, has to decide when to let go of the seat. We can imagine certain cases, plans involving dangerous activities for example, where the planner may never decide to rely completely on the routine behavior. However, in the majority of cases, the learned routine activity can be released from deliberation.

The DS is notified of a routine's capture by monitoring the RSE values of the behaviors. RSE values are set to zero unless the behavior is being primed for activation as part of a sequence. Note that if intentional bias is present, signaling that the sequence is still under deliberative control, the RSE is also set to zero to avoid increasing the AL by double the desired amount. In this case, however, the RSE value that would have been present is still sent to the DS, signaling that the sequence is a captured one.

The intentional bus provides a mechanism for the DS to monitor the status of all schemas. This enables it to observe whether the intended plan sequence did indeed proceed in the required order, ensuring that the outcome is still correct even in the absence of additional intentional bias. It is desirable that the system monitor the progress of the plan for some interval in order to make sure that the sequence has been learned properly and continues to be executed correctly. If it has not, the DS resumes executing the plan through intention. If the routine sequence performs well, however, monitor-

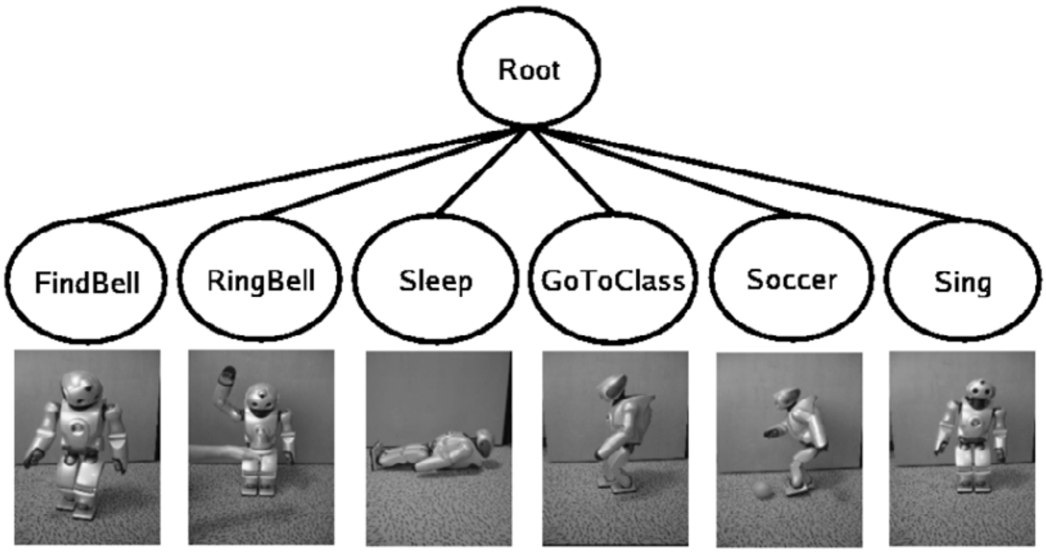

Figure 13 Behavioral tree used for serialization experiments.

ing can be reduced or eliminated entirely. Details about the DS and the implementation of the intentional bias mechanism appear in Ulam and Arkin (2006).

4.3 External Stimuli During Routine Behavior Execution

Because the AL is dependent upon external stimuli, it is possible that an external stimulus will increase the AL of an unrelated behavior during the execution of a routine sequence. For example, if the robot recognizes a human face while executing a soccer sequence, the AL of a dialogue behavior may rise. Whether routine task execution ought to be interrupted by this occurrence should be determined by such factors as the importance of the routine behavior sequence, the relative desire for interaction, and personality preferences of the robot. In our system, the intentional bias mechanism is used to control this process. A series of experiments by Ulam and Arkin (2006) demonstrate the robot's ability to ignore or attend to such distractions based on the strength of the intentional bias signal.

5 Experiments and Results

Several experiments were designed to test the ability of the proposed method to capture and execute routine behaviors. In each experiment, the same behavior sequence was used, which required QRIO to emulate attending a music class. In the music class activity, the robot must go to the class location, locate and play a musical instrument, and sing. All experiments and tasks were performed by the real robot.

The behavior tree used in each experiment can be seen in Figure 13. In addition to the four behaviors that make up the behavior sequence which we wish the robot to learn (*GoToClass*, *FindBell*, *RingBell* and *Sing*), it contains two additional behaviors, *Sleep* and *Soccer*, which compete for activation.

In all experiments, the pair frequency thresholding method was used. The behavior sequence had to be experienced five times, and each pairing had to occur at least 75% of the time for the behavior to be captured. Comparable experimental results have been achieved with the other thresholding approaches and metrics. The likelihood RSE method was used in Experiments 1–3 and the stochastic RSE method in Experiment 4.

5.1 Experiment 1

The goal of the first experiment was to test the system's ability to learn a frequently repeated behavior sequence and to execute it as a routine behavior. The experiment consisted of QRIO executing behaviors under the control of the normal action-selection mechanism. At some intervals, this was interrupted by the DS, which executed the planned music class sequence. For each behavior in the plan, the intentional bias was set to 100, a value selected to be high enough to guarantee that the planned sequence was never interrupted.

Figure 14a provides an overview of the experiment. The left side of the figure shows the plan that was executed by the DS. The right side of the figure lists the contents of the CRL of each behavior upon the completion of the routine capture. Observe that the *GoToClass* behavior, because it is first in the sequence, does not have a trigger schema. Each of the other three behaviors learns its connection to the previous behavior in the plan.

Because the sequence was always executed to completion in a consistent manner, and with no other plans present, each sequence pair was captured at approximately the same time. Upon the completion of the capture, the DS shifted to using an attentional signal instead of deliberative intentional bias. During the next scheduled execution of the music class sequence, the DS initiated the behavior using attention by sending a short burst of intentional bias to the *GoToClass* schema, and then entered a monitoring state for the duration of the sequence's execution. The remainder of the music class sequence was successfully executed entirely through RSE.

Figure 15 compares the AL values of the behavior sequence when executed by the planner and as a learned routine behavior. As the plot lines can be difficult to distinguish, symbols on two bars below each figure are used to indicate the active and primed behaviors for each time segment.

Figure 15a shows the activation values of the system as the planner executes the music class sequence. The robot is initially executing a *Sleep* behavior, which is interrupted by the DS. The planned behaviors are then executed one by one. The overall shape of the AL curves of the sequence closely resembles the idealized curves seen earlier in Figure 11. Upon the completion of the sequence, the robot resumes the *Sleep* behavior.

Figure 15b displays the internal state of the system as the robot executes the music class sequence as a routine behavior. This sequence also begins with the robot sleeping, which is again interrupted by the DS, this time through an attentional instead of an intentional signal. This is characterized by the short spike in the AL of the *GoToClass* behavior. As the *GoTo-*

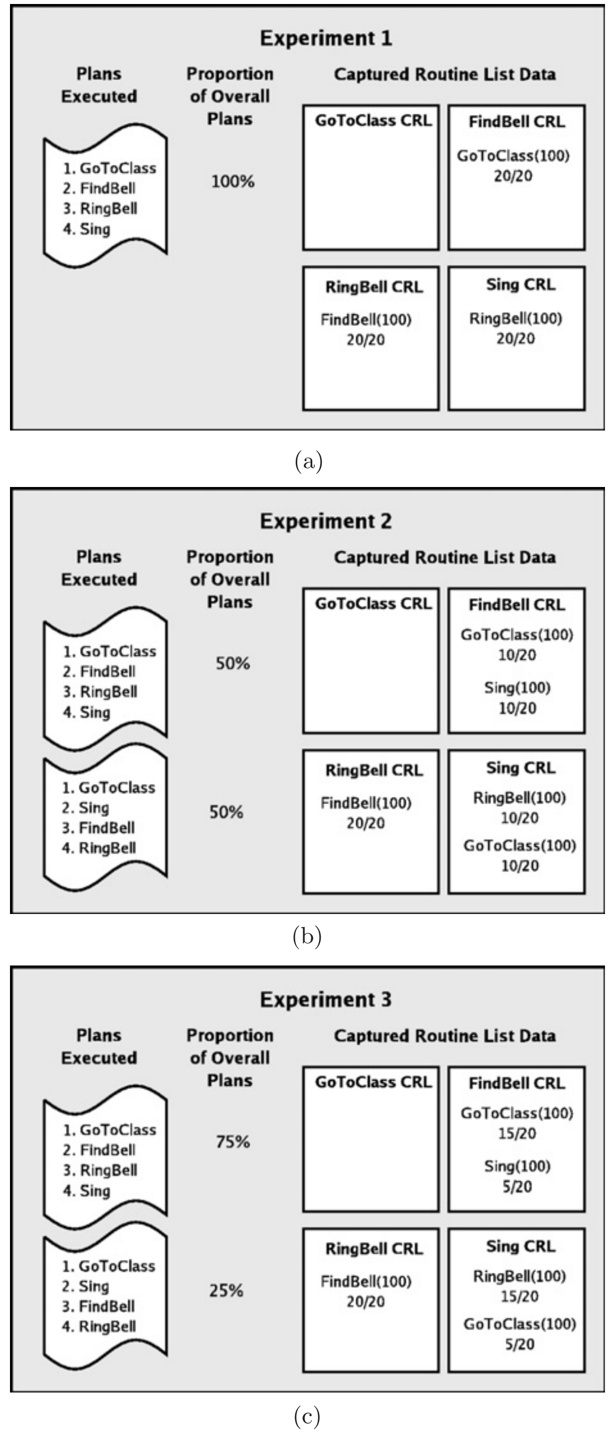

Figure 14 Summary of three experimental setups. Each figure displays the plans executed by the DS, the percentage of time each plan was used, and the internal state of the CRL once the sequence was learned. The CRL of each behavior lists the trigger schemas for that behavior, the average intention value used to activate the behavior, and the fraction of the time the trigger schema is followed by the behavior.

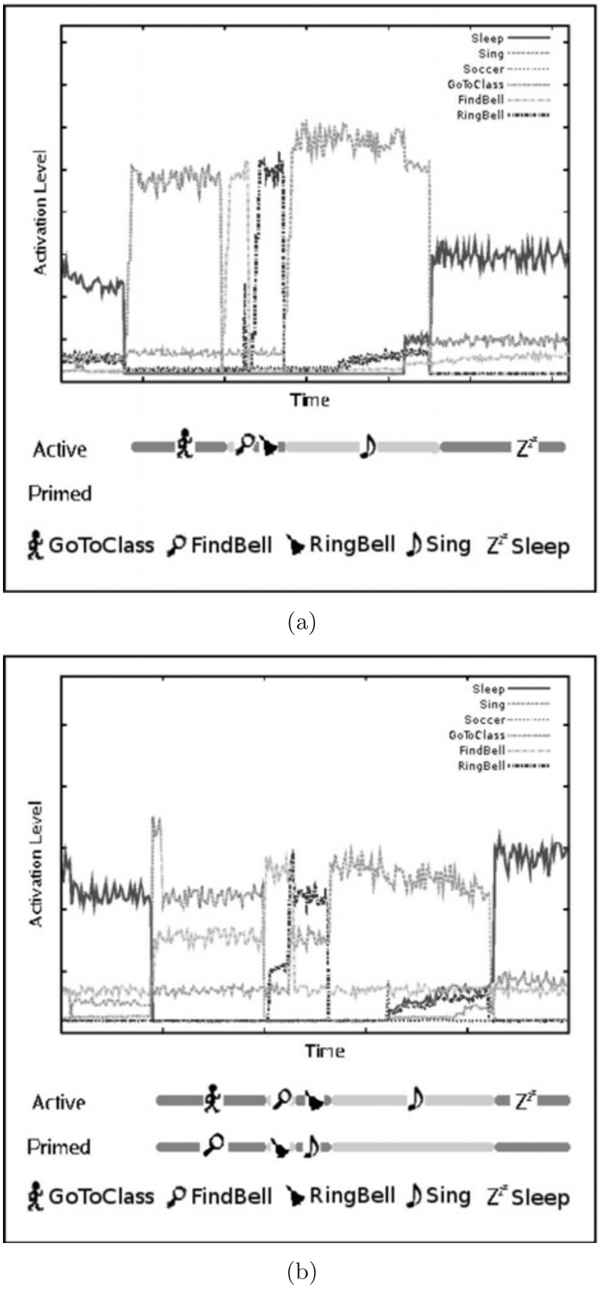

Figure 15 Graphs of the ALs of a sequence of four behaviors executed (a) by the DS and (b) as a routine behavior. AL values of two other competing behaviors are also present for demonstration purposes. Two bars below each graph are used to indicate the active and primed behaviors.

Class behavior becomes active, we see an immediate response from the *FindBell* behavior, whose AL rises as it is primed for activation. As the robot reaches the class and the *GoToClass* behavior completes, *FindBell*

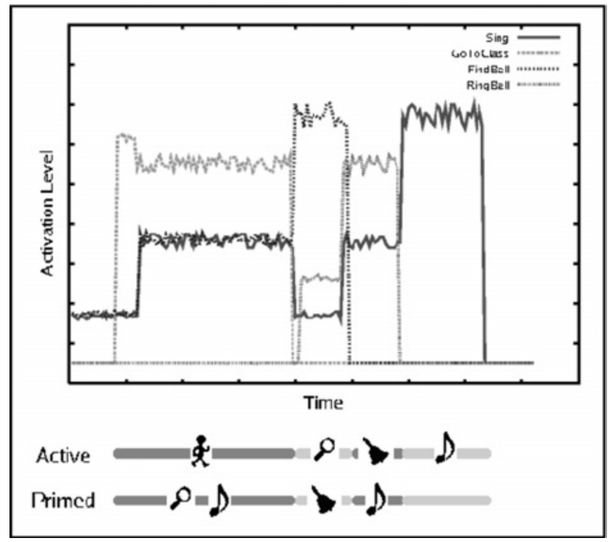

Figure 16 Graph of the ALs of a routine sequence resulting from equal preference variable order plans. Both *FindBell* and *Sing* are primed during the active phase of the *GoToClass* behavior. Both behaviors have equal initial motivation levels and random noise breaks the tie.

is activated and *RingBell* is primed. This results in the step-like graph characteristic of ALs of routine behavior sequences, similar to the idealized graph in Figure 9.

5.2 Experiment 2

The second experiment was performed to test the system's response to variable order plans. Specifically, we are interested in plans that have a fixed number of subtasks that must be achieved, but where the order of some or all of the subtasks is not fixed. These types of plans are fairly common and cover a wide range of activities. In our experiment, we chose to leave the order of QRIO's class activities unspecified. Upon arriving at class, QRIO could choose to sing or ring the bell in either order, although both activities had to be completed.

Figure 14b summarizes the experimental setup. The left side of the figure displays the two plans executed by the planner. In this experiment no preference was given to either plan and they were executed an equal proportion of the time. On the right of the figure, we have another view of the CRL. *GoToClass* again has no trigger schema, and *RingBell* remains unchanged because this behavior always follows after the robot finds the bell. The *FindBell* and *Sing* behaviors now both have two trigger schemas. The *FindBell* behavior will sometimes occur following *GoToClass* and at other times following *Sing*, and the CRL expresses both of these possibilities while also keeping track of the likelihood of each pairing based on previous history.

Figure 16 tracks the AL values over the course of the sequence. Plots of the behaviors not involved in the music class sequence have been omitted for clarity. Initially, the ALs of *FindBell* and *Sing* are very close, separated only by the noise parameter. As *GoToClass* is activated, both behaviors are primed through RSE. Because both plans were equally likely and equal intention levels were used for both behaviors, the RSE values in this case are identical and both behaviors remain close in activation. The noise parameter remains the only separating factor between these two behaviors, and upon the completion of *GoToClass* the *FindBell* behavior is activated. As *FindBell* is always followed by ringing, *RingBell* becomes primed at this time while *Sing* returns to its rest level value. *Sing* is primed again during the *RingBell* behavior and is finally activated as the last behavior in the sequence.

The internal state of the robot, its internal desires or moods, plays an important role in behavior selection and therefore also in sequence execution. Figure 17 presents another set of results from the same experiment where the outcome was affected by the internal state of the robot. In this case, the robot's desire to sing is much greater starting out than its desire to ring the bell. This is apparent from the large difference in ALs of the two behaviors before the sequence begins. As *GoToClass* is activated, both *FindBell* and *Sing* are again primed by the same amount, but the gap between their AL values remains due to the internal state of the robot at this time. This leads to *Sing* being activated upon arriving in class, followed by the bell ringing behavior. This example demonstrates the ability of the system to express the internal state and preferences of the robot.

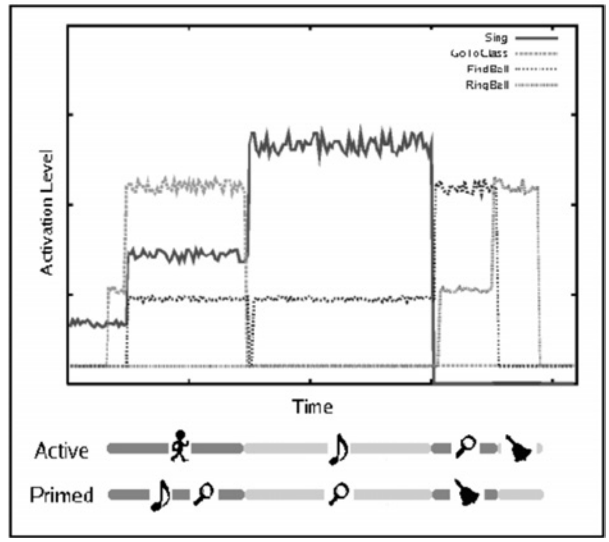

Figure 17 Graph of the ALs of a routine sequence resulting from equal preference variable order plans. Both *FindBell* and *Sing* are primed during the active phase of the *GoToClass* behavior. The *Sing* behavior has a higher initial motivation and is therefore selected.

5.3 Experiment 3

The third experiment was based on the same setup as Experiment 2, but one of the plans was given preference over the other. As can be seen in Figure 14c, executing bell ringing before singing was preferred and 75% of the executed plans used this order. The CRL, which tracks the likelihood of each pairing occurring, reflects this preference in the *FindBell* and *Sing* behaviors.

Figure 18 shows the AL plot for this sequence. Initially, the AL values for *FindBell* and *Sing* are again very close. Just as in the previous experiment, as the *GoToClass* behavior becomes activated, both *FindBell* and *Sing* are primed. However, under the pair frequency thresholding method the RSE is scaled by the expected probability of a pairing occurring, which results in different levels of excitation for the two behaviors. As *FindBell* is more likely to follow *GoToClass*, its excitation level is greater. This results in the bell ringing behavior being executed before singing.

It is important to note the amount of separation between the primed AL values of *FindBell* and *Sing*, and to compare this to the separation at the beginning of Figure 17. In Figure 17 there is a strong internal desire associated with the *Sing* behavior but not with *FindBell*. This causes a large separation in the AL values of the two behaviors, which is greater than the separation due to RSE in Figure 18. This leads to the conclusion that a preference for one plan ordering over another will indeed bias the behavior selection towards preferring that order, but without eliminating

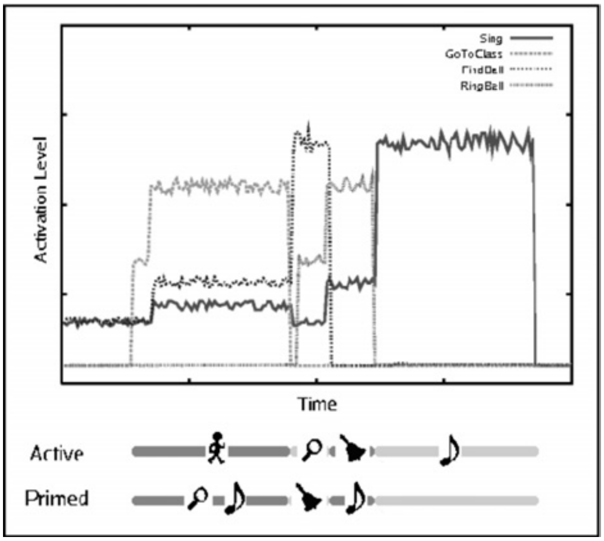

Figure 18 Graph of the ALs of a routine sequence resulting from biased preference variable order plans. Both *FindBell* and *Sing* are primed during the active phase of the *GoToClass* behavior. The preferred behavior, *FindBell*, has a higher RSE value, causing it to be selected.

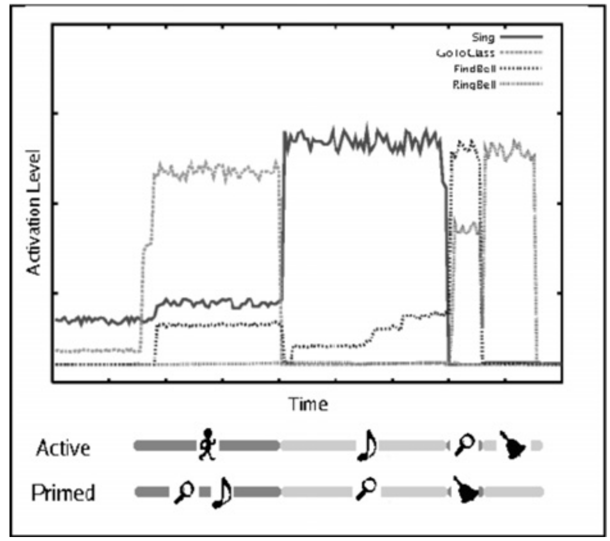

Figure 19 Graph of the ALs of a routine sequence resulting from biased preference variable order plans. Both *FindBell* and *Sing* are primed during the active phase of the *GoToClass* behavior. The *Sing* behavior has a higher initial motivation, and although it is not the preferred behavior, its overall AL exceeds that of the *FindBell* behavior.

the occurrence of the less preferred sequence. If QRIO again experiences a strong desire to *Sing* and little or no desire for *FindBell*, then, as can be seen in Figure 19, the less preferred behavior sequence will indeed occur.

5.4 Experiment 4

The final experiment utilizes the routine sequence captured in Experiment 3, but the RSE is calculated using the stochastic instead of the likelihood method. The stochastic method aims to generate a distribution of AL values that resembles the plan or pairing distribution. Specifically, in the case where the initial AL values of two behaviors are very close, the goal is to ensure that each behavior has a probability of being activated that is proportional to the likelihood of the pairing.

The resulting AL graph can be seen in Figure 20. The most notable difference occurs during the time when the *GoToClass* behavior is active. In this case, *Sing* is the less likely behavior, but its AL exceeds that of *FindBell* approximately 28% of the time, roughly proportional to the likelihood of the *GoToClass–Sing* pairing. This ensures that all else being equal, all pairings will have a probability of being selected that is proportional to the likelihood of the pairing as observed under the control of the DS. Of course, when

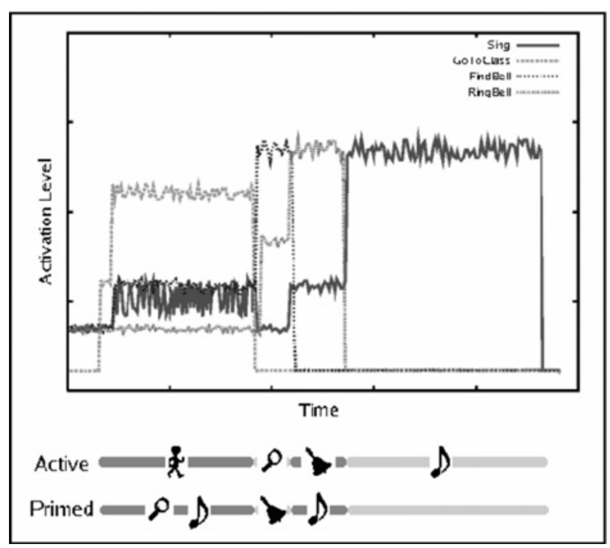

Figure 20 Graph of the ALs of a routine sequence resulting from biased preference variable order plans, demonstrating the stochastic RSE calculation method.

there is a preference for one activity over the other as a result of the internal state, this will play a strong role in behavior selection, as previously demonstrated.

6 Discussion and Conclusions

In this article, we have demonstrated a method by which sequential tasks executed through deliberation can become routine, such that the mechanism of their execution is shifted to the reactive behavior level and deliberative planning is no longer involved in the execution of the sequence. This allows the separation of routine and non-routine activities, while reducing the load on the planner and freeing system resources.

The experimental results presented above demonstrate the flexibility of the routine capture system, as well as many of its strengths. This method allows an unlimited number of behavior pairings to be captured into routines, enabling the robot to learn and perform sequenced tasks in a natural and ordered manner. Numerous behaviors can be linked to each trigger schema, allowing the same behavior to be reused in many plans. Which behavior is selected for activation depends not only on the observed experiences, but also on the current internal state of the robot and the external stimuli. As a result, the routine behavior execution builds upon the robot's ability to express its desires instead of suppressing it. Over time, this leads to more interesting behavior combinations that vary depending on the robot's environment, a key property for long-term robot interaction.

Several interesting and important extensions to this model are left to consider. One natural extension to examine is using a richer and more flexible trigger schema representation. Whereas the current algorithm only looks one step back in the history, it could be extended to trace further back along the sequence of past behaviors. This would enable the system to differentiate between the sequences $A \rightarrow B \rightarrow C$ and $D \rightarrow B \rightarrow E$, where just knowing that B is active does not provide enough information to determine whether E or C is more appropriate as the next behavior. In this case, the ambiguity can be eliminated by checking further back in the history. This extension would result in more accurate and reliable execution of the routine behavior sequences.

Another question of interest is whether the robot should be allowed to unlearn, or forget, previously captured routines. Such an extension seems natural as the environmental features or the robot's tasks may permanently change, resulting in some learned sequences becoming useless or inappropriate. The implementation and analysis of this mechanism has been left for future work.

While many directions are left to be explored, the system presented here is already the first of its kind to autonomously perform the capture and execution of higher-level behavior sequences. The implementation and execution of this complete system on an autonomous humanoid robot brings us a step closer to achieving the goal of natural and varied long-term human–robot interaction.

Acknowledgments

This research was conducted while the authors were visiting the Sony Intelligence Dynamics Laboratories (SIDL). We would like to thank Kazumi Aoyama, Hideki Shimomura, and the other members of the SIDL development team for their invaluable help with this project. We also thank Masahiro Fujita and Toshitada Doi for making this collaboration possible.

References

Arkin, R., Fujita, M., Takagi, T., & Hasegawa, R. (2003). An ethological basis for human–robot interaction. *Robotics and Autonomous Systems*, *42*(3–4), 191–201.

Badgalyan, R. (2000). Executive control, willed actions, and non-conscious processing. *Human Brain Mapping*, *9*(38), 38–41.

Cisek, P. (2001). Embodiment is all in the head. *Behavioral and Brain Sciences*, *24*(1), 36–38.

Cleeremans, A. (1993). *Mechanisms of implicit learning: Connectionist models of sequence processing*. Cambridge, MA: MIT Press.

Cooper, R. P., & Glasspool, D. W. (2001). Learning action affordances and action schemas. In R. M. French & J. P. Sougne (Eds.), *Perspectives in neural computing* (pp. 133–142). Berlin: Springer.

Cooper, R., & Shallice, T. (1997). Modeling the selection of routine action: Exploring the criticality of parameter values. In *Proceedings of the 19th Annual Conference of the Cognitive Science Society* (pp. 130–135). Stanford University, CA.

Cooper, R., & Shallice, T. (2000). Contention scheduling and the control of routine activities. *Cognitive Neuropsychology*, *17*(4), 297–338.

Cooper, R., Shallice, T., & Farringdon, J. (1995). Symbolic and continuous processes in the automatic selection of actions. *Hybrid Problems, Hybrid Solutions*, *2*, 27–37.

Fujita, M., Kuroki, Y., Ishida, T., & Doi, T. (2003). Autonomous behavior control architecture of entertainment humanoid robot SDR-4X. In *Proceedings of the IEEE/RSJ international Conference on Intelligent Robots and Systems*. Las Vegas, NV.

Garforth, J., Meehan, A., & McHale, S. (2000). Attentional behavior in robotic agents. In *Proceedings of the International Workshop on Recent Advances in Mobile Robots*. Leicester, UK.

Hoshino, Y., Takagi, T., Profio, U., & Fujita, M. (2004). Behavior description and control using behavior module for personal robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*. New Orleans, LA.

Houghton, G., & Hartley, T. (1995). Parallel models of serial behaviour: Lashley revisited. *Psyche*, *2*(25).

Keele, S., Ivry, R., Mayr, E., Hazeltine, E., & Heuer, H. (2004). The cognitive and neural architecture of sequence representation. *Psychological Review*, *110*, 316–339.

Lashley, K. S. (1951). The problem of serial order in behavior. In L. Jeffress (Ed.), *Cerebral mechanisms in behavior*. New York: Wiley.

Norman, D. A., & Shallice, T. (1986). Consciousness and self-regulation: Advances in theory and research. In D. S. R. Davidson & G. Schwartz (Eds.), *Attention to action: willed and automatic control of behavior* (Vol. 4, pp. 515–549). New York: Academic Press.

Prescott, T., Gurney, K., & Redgrave, P. (2003). Basal ganglia. In M. A. Arbib (Ed.), *The handbook of brain theory and neural networks* (2nd ed.). Cambridge, MA: MIT Press.

Sawada, T., Takagi, T., & Fujita, M. (2004a). Behavior selection and motion modulation in emotionally grounded architecture for QRIO SDR-4X II. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Sendai, Japan.

Sawada, T., Takagi, T., Hoshino, Y., & Fujita, M. (2004b). Learning behavior selection through interaction based on emotionally grounded symbol concept. In *Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots*. Los Angeles, CA.

Sun, R., & Giles, L. (2001). *Sequence learning: Paradigms, algorithms, and applications*. Berlin: Springer.

Tanaka, F., Noda, K., Sawada, T., & Fujita, M. (2004). Associated emotion and its expression in an entertainment robot QRIO. In *Proceedings of the International Conference on Entertainment Computing* (pp. 499–504). Eindhoven, the Netherlands.

Ulam, P., & Arkin, R. (2006). Biasing behavioral activation with intent (Technical Report No. GIT-GVU-06-11). Graphics, Visualization, and Usability (GVU) Center, Georgia Institute of Technology.

About the Authors

Sonia Chernova is a Ph.D. student in the Computer Science Department at Carnegie Mellon University. She received her undergraduate degree in computer science and robotics from Carnegie Mellon University in 2003. Her research interests include learning, interaction and teaching in robotic systems. From July to October 2005 she was a visiting researcher at the SIDL.

Ronald C. Arkin holds the rank of Regents' Professor and is the Director of the Mobile Robot Laboratory in the College of Computing at Georgia Institute of Technology. From June to September 2005, Professor Arkin held a Sabbatical Chair at the SIDL. His research interests include behavior-based reactive control and action-oriented perception for mobile robots and unmanned aerial vehicles, hybrid deliberative/reactive software architectures, robot survivability, multi-agent robotic systems, biorobotics, human–robot interaction, robot ethics, and learning in autonomous systems. He has over 130 technical publications in these areas. He was elected a Fellow of the Institute of Electrical and Electronics Engineers (IEEE) in 2003, and is a member of the American Association for Artificial Intelligence (AAAI) and the Association for Computing Machinery (ACM).

E-mail: arkin@cc.gatech.edu