Biasing Behavioral Activation with Intent

Patrick Ulam and Ronald Arkin
College of Computing
Georgia Institute of Technology
Atlanta, USA 30332
Email: pulam, arkin@cc.gatech.edu

Abstract— **Deliberate control of an entertainment robot presents a special problem in balancing the requirement for intentional behavior with the existing mechanisms for autonomous action selection. We propose that the intentional biasing of activation in lower-level reactive behaviors is the proper mechanism for realizing such deliberative action. In addition, we suggest that directed intentional bias can result in goal-oriented behavior without subsuming the underlying action selection used to generate natural behavior. This objective is realized through a structure called the intentional bus. The intentional bus serves as the interface between deliberative and reactive control by realizing high-level goals through the modulation of intentional signals sent to the reactive layer. A deliberative architecture that uses the intentional bus to realize planned behavior is described. In addition, it is shown how the intentional bus framework can be expanded to support the serialization of planned behavior by shifting from direct intentional influence for plan execution to attentional triggering of a learned action sequence. Finally, an implementation of this architecture, developed and tested on Sony's humanoid robot QRIO, is described.**

Index Terms— **robotics, behavior-based robotics, deliberative control, reactive-deliberative control**

I. Introduction

As robots' interactions with humans become increasingly prevalent both in a service capacity [1] and in personal interaction roles [2], it becomes important to address the problem of incorporating intentional action into these systems in a natural way. This is an exceptionally challenging proposition for entertainment robots because the intentional control needs to provide for goal directed action without subsuming the existing action selection mechanisms used to provide the robot with "personality" [3]. In the case of a humanoid, such as Sony's QRIO shown in figure 1, the problem is further compounded by the expectations placed upon anthropomorphic robots by the user. In order for a humanoid's actions to be perceived as natural, its deliberative capabilities need to extend beyond that of simple plan execution and move closer to the range of deliberative capabilities afforded to humans.

While deliberation can mean many things, both in humans and in robots, it is common for psychologists [4] to place tasks requiring deliberative control in humans into several categories:

1) Tasks involving planning or decision making
2) Tasks involving troubleshooting
3) Tasks containing novel sequences of actions
4) Tasks that are dangerous or difficult
5) Tasks that require the overriding of habitual responses

While intelligent systems may be able to benefit from handling all the deliberative tasks described above, the three most important, at least from the viewpoint of an entertainment robot, are tasks requiring planning, tasks requiring novel sequences of actions, and tasks requiring the overriding of habitual responses. For example, a task that may be asked of an entertainment robot is the delivery of a newspaper to the user. This can be formulated as a typical planning activity in which a plan is generated to find, collect, and deliver a newspaper at a particular time. In fact, many such systems have be created to generate and execute such plans [5][6]. For an entertainment robot, however, often additional actions beyond those merely necessary to achieve this plan are required if its behavior is to be perceived as natural. For example, if someone tries to attract the robot's attention for interaction while executing this plan, it should interact with that person unless the task is of very high priority.

Planning also serves the purpose of providing a novel source of actions for the robot to perform. This is especially important for a robot that is to provide long-term interaction with a human, as this can result in the previously unseen behavior that keeps the robot's activities non-boring from the user's perspective. In addition, these novel action sequences can occur without requiring additions to its behavioral repertoire.

Another important deliberative task is the suppression of behavior. For humans, it is just as important to know when it is appropriate to perform an action such as calling a friend, as it is to know when not to (e.g., in a business meeting). For an entertainment robot, suppression of a behavior may be the result of a request to remain quiet when it knows the user is watching a movie. In this case, dialog-oriented behaviors should be suppressed even if the robot maintains a desire to interact. Such a user request should result in the robot being highly unlikely to interact, but not eliminate its ability to do so entirely if an important piece of information is needed to be communicated to the user. Therefore, in order to provide for natural behavior, the deliberative system should be able to not only activate appropriate actions but also to suppress inappropriate actions.

This work proposes that intentional bias of the action selection mechanism used by entertainment robots such as QRIO is the proper means by which to express such deliberate control in a natural manner. Because of the delicate balance required between a robot's behavior serving in an entertainment and service capacity, it is inappropriate for these functions to exist in isolation. One does not want a robot meant to serve as

Fig. 1. Sony's entertainment robot QRIO

a companion to lose all semblance of personality or related entertainment characteristics when it is pursuing high-level goals. Similarly, it is not desirable for an entertainment robot to exist solely in the context of satisfying the immediate needs of the robot or user in a purely reactive manner. In order to achieve the types of deliberate control described above, it is forwarded that a fine-grained application of intention is needed if the behavior is to be perceived as natural.

An example of where such fine-grained deliberate control may result in more natural behavior might be found in an instance where a user of an entertainment robot has lost an object, for example a valuable ring. The user asks the robot to help find this ring. In this scenario the robot may produce a plan to explore a particular room looking for the object. The user probably doesn't want the robot talking about the weather at this point but instead wants the robot to perform this task without reverting to less important behaviors such as interaction or play. In this case the robot should execute the plan regardless of any extraneous motivations to perform alternate actions.

In another scenario, the user may ask the robot to retrieve loose coins that may have been accidentally dropped. A similar plan to search the room could be generated. In this case, however, the request is not as important to the user. Thus the robot may only search for change when there are no more important goals to fulfill. Even when executing other actions or other plans, however, the opportunistic sight of fallen change should increase the likelihood that the robot starts executing a coin retrieval plan. Conversely, if the robot is searching for the coins, it should be able to be distracted from the task, if more desirable or timely actions become possible.

Our research presents a mechanism for providing uniform intentional control of the types described above. This mechanism, called the intentional bus, serves as a gateway between the reactive and deliberative layers in the architecture. The intentional bus converts high-level goal-oriented tasks generated by the deliberative system into intentional signals that serve to bias behavioral activation in the reactive layer. These biases serve as a means of directing the robot towards accomplishing the goals of the deliberative system without overriding the existing action level computations used to generate natural behavior. In addition, we show that this interface is general enough to provide interaction beyond that of intentional bias by expanding it to handle attentional triggering of learned sequences of behavior. Finally, we present an implementation and results for this form of deliberative control on Sony's humanoid QRIO.

The remainder of the paper is composed as follows: Section II provides an overview of related work on hybrid reactive-deliberative architectures, psychological evidence for intentional bias, and an overview of the action-selection architecture for the entertainment robot QRIO used in this work. Section III describes the intentional bus developed for this research and its interactions with the deliberative and reactive architectural layers. Section IV extends the intentional bus design to allow for attentional triggering of learned serial actions. Section V describes the deliberative architecture developed using intentional bias to provide high-level control for the entertainment robot QRIO. Section VI provides an evaluation of the use of intentional bias for control of QRIO. Finally, section VII concludes the paper and presents an agenda for future work.

II. Related Work

A. Hybrid Reactive-Deliberative Architectures

Hybrid reactive-deliberative systems have been successfully deployed for many years, serving as a mechanism for combatting the shortfalls of purely reactive and purely deliberative architectures. Arkin and Gat provide excellent overviews of a number of hybrid architectures in [7] and [8]. In these architectures, the deliberative layer is usually coupled to the reactive layer in one of three manners: as a mechanism for generating actions at a differing time scale than the reactive layer; as a means to guide reaction; and through the direct coupling of reaction and deliberation, each guiding the other [9]. In most cases, the role of the deliberative system is to plan which actions should execute at a particular time to achieve a goal. In many cases, the interaction between the deliberative layer and the reactive layer is mediated by a sequencing layer which selects the portions of the plan that should execute at a given time. The sequencer then activates the behaviors required to realize the next step in the plan.

In most architectures, this selection is a winner-take-all process in which the behavior chosen is executed if the environmental state matches that expected by the sequencer. This activation is almost always a binary activation without support for increased or decreased likelihood of behavioral activation. Some work has investigated alternative means of combining deliberative information into the reactive layer. Payton [10], as well as Wagner and Arkin [11], have looked at means of incorporating planned navigational information directly into the behavioral controller. These plans provided

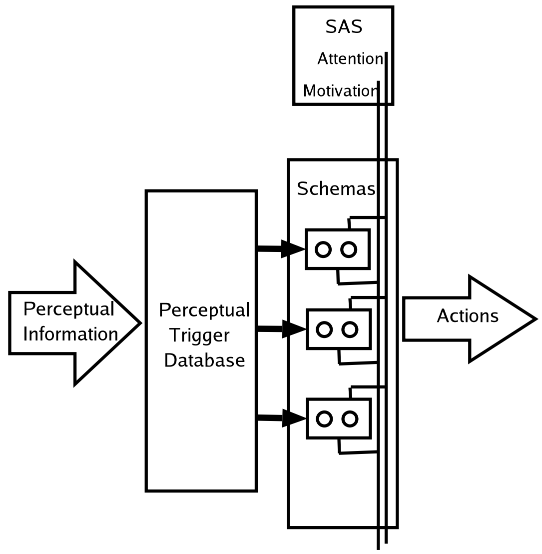


Fig. 2. Norman and Shallice's Model of the role of the SAS [4]

a navigational bias into a reactive controller, which served as a means of influencing the expression of the currently executing behavioral assemblage while still allowing other reactive behaviors, such as obstacle avoidance, to be active. The plans used in their work were not incorporated into the behavioral selection process itself, however.

Numerous other means of incorporating deliberative information into the reactive layer have been investigated. A thorough review of this area appears in [7]. A few representative examples are now discussed. The Motivated Behavioral Architecture [12] was developed to decouple deliberate control for various navigation tasks. Planning in this architecture, when it is available, was used to select which subtasks should be addressed by the behavioral modules present in the system. Rosenblatt's DAMN architecture [13] provides a mechanism in which behaviors in the traditional sense, as well as planners instantiated as behaviors, cast votes as to the appropriate action the robot should take. The arbiter in this architecture combines these votes to select the appropriate action to take. Significant challenges exist in specifying voting semantics making arbitration between different behaviors and fine-grained deliberative control difficult, however. Carpenter *et. al.* developed an architecture for robot soccer in which deliberate advice from a coach is integrated via the addition of suggested behaviors to the set of executable behaviors [14]. The effect of this advice, however, is highly dependent on the scoring function used by the behavioral arbiter used and appears unable to be influence behavioral selection in a non-binary manner.

B. A Psychological Basis for Deliberate Control of Reactive Behaviors

One particularly compelling model for the deliberative control of automatic actions in humans, developed by Norman and Shallice, is the supervisory attentional system (SAS) [4]. Created to account for lapses of action in routine activity, they propose the SAS acts as the controller over the expression of automatic behaviors within humans. An overview of the interaction of the SAS with the underlying behavioral layer can be seen in figure 2. Their overall behavioral model can be divided into two major interacting components, the portion containing basic well-known actions called the contention scheduling mechanism, and the portion which provides deliberate influence over action which is the supervisory attentional system.

The contention scheduler involves multiple sets of simple, well defined actions called schemas. Schemas can be atomic or a well-defined series of schemas. A set of schemas is activated when perceptual information in the environment matches certain triggering conditions found in a perceptual feature database. The strength of the activation is directly related to the degree of match between those features and features in the database. While it is beyond the scope of this paper to discuss contention scheduling in detail, there has been significant work in formalizing models of contention scheduling [15] [16] as well as evaluating its feasibility as an action-selection mechanism for robots [17] [18].

The contention scheduling system serves as a means of providing reactive control for routine actions. In the case of novel sequences of actions that are not known *a priori*, the contention scheduling system alone is insufficient to express such behavior. Norman and Shallice [4] propose that the SAS is the component that provides the additional source of control needed to achieve such novel actions. They suggest that the SAS in humans interacts with the reactive contention scheduling layer through the use of motivational and attentional threads. The attentional thread represents a bias to the activation levels of the sets of schema found in the contention scheduler resulting in an excitation or inhibition of a given schema, but never activates the schema directly. The motivational thread provides a similar mechanism to operate over long term goals. These biases can allow for the activation of novel sequences of behavior even if that sequence did not already exist in the contention scheduler. How the SAS actually does this is to date still ill-defined, although Shallice and Burgess have examined possible mechanisms by which the SAS may generate and evaluate plans [19]. They do not, however, talk about the influence of the attentional and motivational threads with the lower-level reactive layer in much detail.

While it has been proposed [7] that Norman and Shallice's model of willed behavior may provide a suitable guide for integration of reactive and deliberative control in robotic systems, little has been done to investigate such mechanisms. Garforth *et. al.* appear to have been alone in doing so thus far. They looked at using neural mechanisms to mimic processes of the supervisory attentional system for a simulated robot [20] [21], investigating the suppression of a particular behavior in a simple foraging task using a previously trained neural network. The controller in this architecture appears to be limited to the suppression of a particular behavior and cannot generate novel behavioral sequences itself. In addition, their architecture does not appear to support differing values of suppression beyond binary suppression. They also discuss a mechanism for learning new automatic actions through the use of an

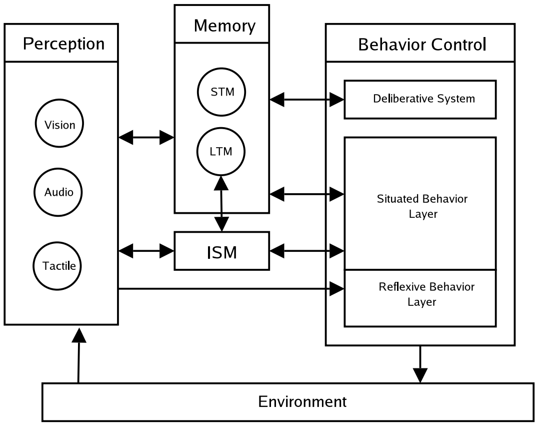


Fig. 3. Overview of the EGO architecture

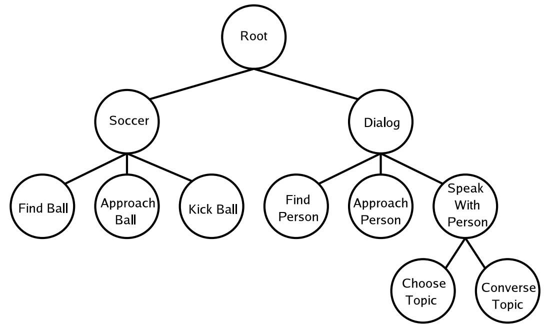


Fig. 4. Example behavioral tree in the EGO architecture containing two main behaviors: Soccer and Dialog

episodic memory triggered by SAS activity. Unfortunately, the mechanisms by which this is implemented and utilized are unclear, nor do they come anywhere close to addressing the level of deliberative task complexity required by a humanoid entertainment robot such as QRIO.

C. Action Selection and Behavioral Representation in the EGO Architecture

Action selection, the problem of determining which action to perform among many possible conflicting actions, has been studied extensively in both the agents and robotics communities. Numerous models have been produced including ethologically-guided [22] and spreading activation models[23]. Less effort, however, has been spent on considering the role of intention in action selection. For example, Terzoupoulos *et. al.* has looked at using intention to provide goal-oriented behavior for simulated fish [24]. Intention in his work, however, is not deliberative in the sense of the research described in this article, and rather serves as a binary action selection mechanism with which the system selects different behaviors such as feeding or escaping based on various behavioral parameters such as hunger or fear. An overview and comparison of a number of alternate action selection mechanisms can be found in [25].

For this work, however, we use the action selection mechanism incorporated within the EGO architecture [26]. The EGO architecture, or Emotionally GrOunded architecture, utilizes the OPEN-R framework originally having its roots in the design of the quadruped robot AIBO and later expanded to the humanoid QRIO, the research platform for this work. An overview of the components within the EGO architecture is shown in figure 3. In the EGO architecture, behaviors are organized in a tree structure according to their conceptual relationship with each other [27]. For example, a high-level soccer behavior may be decomposed into several low-level behaviors such as finding and kicking a ball.

The EGO architecture currently uses two situated behavioral layers (SBL), namely the normal situated behavior layer (NSBL) and the reflexive situated behavioral layer (RSBL). Each SBL provides mechanisms for the concurrent evaluation, concurrent execution, and preemption of the various behaviors running on QRIO. An example of a very small NSBL tree appears in figure 4. In the EGO architecture, behaviors maintain an activation level and set of resources associated with them. Every cycle, most behaviors calculate their activation levels after which a selection process occurs. In the action selection process, the behavior with the greatest activation level is selected for execution for which there are available resources. If there are remaining resources, this process is repeated until there are no more free resources. This allows for the concurrent execution of non-conflicting behaviors to occur.

In the EGO architecture, the activation level calculation is a result of an ethologically-based homeostatic action selection mechanism [27]. The homeostatic control module in QRIO attempts to keep a number of internal state variables, such as desires for interaction, rest, and activity, within certain bounds. By evaluating the levels of these particular variables, as well as external stimuli, actions can be selected by increasing the activation of behaviors that will satisfy the robot's internal drives. The further a particular internal state variables is out of the specified range, the greater its influence on action selection. For example, a behavior that causes QRIO to sit on the floor quietly for a period of time may satisfy QRIO's internal variable for rest. The higher QRIO's desire for rest is, the higher the activation value is for the sit behavior. The effects of the internal state variables on action selection are coupled with the additional influence of various external releasing mechanisms found in the world. For example, even if QRIO's internal state variable corresponding to interaction is within the proper bounds, if QRIO sees a face, the activation level of interaction oriented behaviors will be increased.

More precisely, the activation level for the behaviors in the EGO architecture are calculated as:

$$B_v = \beta M_v + (1-\beta) R_v, \tag{1}$$

where B_v is the activation level, M_v is the motivational value based on the robots internal state, and R_v is the releasing value. The releasing value is be specified as:

$$R_v = \alpha \Delta S + (1-\alpha)(S + \Delta S), \tag{2}$$

where S is the current satisfaction as measured by the internal state variables and ΔS is the expected change is satisfaction if the particular behavior were to become active. A detailed

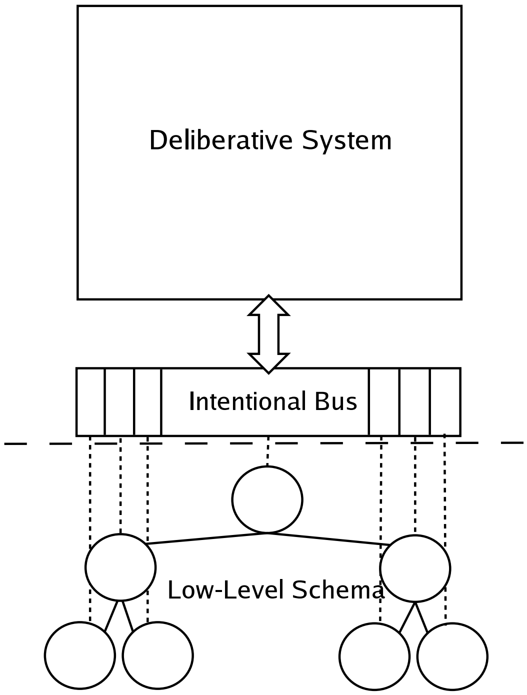


Fig. 5. Overview of the interaction between the deliberative system, intentional bus, and lower level reactive behavior

Schema Name	Dance	Sing	......	Approach	Greet
State	Running	Stopped	......	Stopped	Stopped
Completion	False	False	......	True	False
Activation	100	50	......	10	10
Intention	60	0	......	0	0
Attentional Flag	False	False	......	False	False
Process ID	1	2		22	23

Fig. 6. The state information contained within the intentional bus with sample values

discussion on the activation level computation and action selection within the EGO architecture can be found in [28].

The action selection mechanism in the EGO architecture bears some resemblance to the contention scheduling model of Norman and Shallice ,where the situated behaviors in the EGO architecture correspond to the reactive schema within the contention scheduling model. In both the EGO architecture and the contention scheduling model, top-down activation of situated behaviors occur. In addition, both the EGO architecture and the contention scheduling model provide environmental triggering of low level behaviors through the use of releasing mechanisms. The major difference between the two action selection mechanisms is the lack of lateral inhibition within the EGO architecture.

To reiterate, our hypothesis is that through the intentional bias of activation levels such as described here, goal-oriented behavior can be added to the system without compromising the strength of the existing action selection mechanism.

III. The Intentional Bus as a Mechanism for Deliberative Behavioral Biasing

In order to combine the execution of reactive behavior with goal-oriented behavior in a natural manner, a mechanism for providing, monitoring, and maintaining intentional control is necessary. The intentional bus is the component that provides these services and allows the deliberative system to interact with the reactive behaviors in a coherent manner. An overview of the interaction between the deliberative layer and the lower level behavioral layer appears in Figure 5. The intentional bus provides three major functions:

1) Monitoring and reporting the status of the underlying schema.
2) Biasing the activation levels in the reactive layer via intentional threads.
3) Maintenance of the intentional bias in response to changing activation levels.

The intentional bus serves as a repository of information about the underlying behavioral level for use by a deliberative system. In addition, this state information is used by the intentional bus itself for calculating the appropriate intentional bias to send to schemas. The bus stores information pertaining to the state of all schemas (running, stopped, etc.), information pertaining to the recent completion of all schemas, activation levels of all schemas, intentional levels of all schemas, and the process IDs of all schemas. This information is derived from two different locations: Intentional data comes from the deliberative layer while all other information is received from the reactive layer. An example of the state information stored and utilized by the intentional bus is depicted in figure 6. In addition, figure 7 shows the origins and destinations for all data in the bus as well as the computations performed on that data by the bus.

The second function of the intentional bus, the biasing of activation levels for a schema, is initiated when a request from the deliberative system is received. This request is composed of three parts:

1) s: the schema to bias
2) m: the magnitude of the bias
3) a: an attentional flag, set to indicate if the bias should be an attentional trigger (described in detail in the next section)

The magnitude of the bias is calculated by treating m as a percent of the total activation of all schema, such that,

$$b = \frac{m \sum a_i}{100}, \tag{3}$$

where a_i represents activation level for schema i. Other means of calculating the bias can be substituted without loss of generality, however. The resulting bias value is placed on the bus and is sent to the appropriate schema, s, via the intentional threads. This bias is added to the current activation level by the schema resulting in the level increasing or decreasing by the specified amount. The magnitude of the bias is stored along with the other additional state information about the schema for intention level maintenance. It is through this

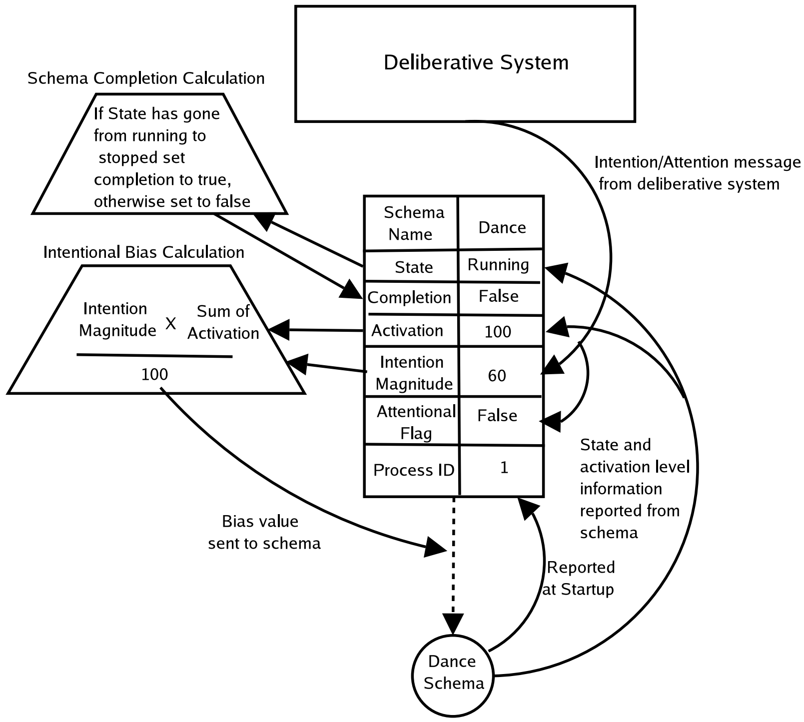


Fig. 7. Dataflow in the intentional bus. Arrows show origin and destination of the data. Trapezoids show computations internal to the intentional bus.

intentional bias of behavior that goal-directed behavior can occur. The bias on the activation level of the behavior can have a large number of effects ranging from certain execution for values of $m > 100$, slight influence towards activation when $0 \leq m \leq 100$, to inhibition of the behavior when $m \leq 0$.

The final function of the intentional bus is the active maintenance of the intentional signals. As activation levels change in the system due to the termination and activation of schema as well as the detection of releasing mechanisms in the environment, the intentional bias currently applied to a schema may no longer remain at the appropriate level. In the event there is a change in the activation levels of the underlying schema, the intentional bus uses the stored values of m and recalculates the intentional bias for each schema. The new bias is sent to the schema and in this way ensures the desired level of bias remains applied to a schema.

Figure 8 shows an example of intentional bias and modulation with two behaviors. Behavior one remains active until the intentional bus sends an intentional signal to behavior two, raising its activation level higher than behavior one. While executing behavior two, the releasing mechanism corresponding to behavior one is detected in the environment, resulting in its activation level increasing. In this case, the intentional bus detects this change and actively maintains the intentional bias to behavior two.

A second example of intentional bias that does not override the existing action selection can be seen in Figure 9. In this example, four different behaviors are present with behavior one currently running. The intentional bus receives a request that behavior two be biased with a moderate magnitude. Behavior two's activation level increases, but does not exceed that of the currently executing behavior. When behavior one completes, however, the bias is great enough for behavior two to become active.

IV. The Intentional to Attentional Shift for Routine Activity

One important capability for a robot that is designed for long-term interaction is the ability to learn routine activities from repeated intentional execution of plans. Routine activities in the context of this work are plans generated by the deliberative layer that are repeated many times. For instance, if the robot were to retrieve a newspaper on a daily basis for the user, the plan generated for newspaper retrieval would be a candidate for a routine activity. When a previously planned activity becomes a routine activity the explicit plan execution and monitoring that occurs in the deliberative system can migrate to the lower-level action selection mechanism, freeing the resources that formerly had been dedicated to generating and executing such plans. We propose that this transition can occur via the use of an attentional signal to activate a learned behavioral sequence. Instead of the deliberative system executing an activity through the intentional bias of the underlying behaviors, an attentional signal is instead sent to the first behavior in the sequence in order to start a chain-activation of the requisite schemas. This notion of initiating sequences of actions at a low-level via a high-level process has been touched upon by Bonasso and Kortenkamp [29] but never investigated fully.

Describing the specific learning mechanism used within the reactive layer for learning serial actions is beyond the scope of this paper, but details can be found in [30]. Other alternative designs are also possible. For the purposes of this article, it is assumed that such a mechanism exists and that the reactive

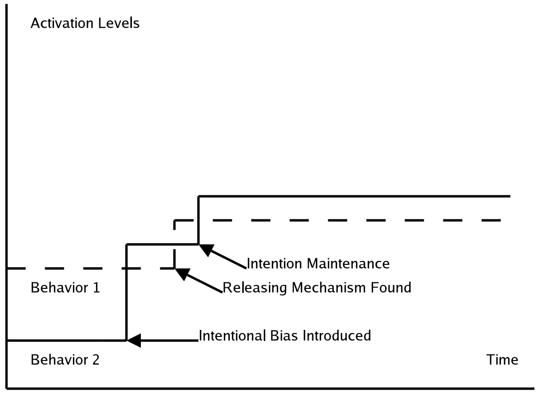


Fig. 8. An idealized graph of activation values for two behaviors showing how intention can bias the activation level of a behavior and how the intentional bus modulates the intention in response to outside influences. In this example, behavior two is biased via intention enough to cause it to run. While behavior two is executing, however, the releasing mechanism for behavior one is found which causes its activation level to increase. The intentional bus observes this and adjusts behavior two's bias appropriately.

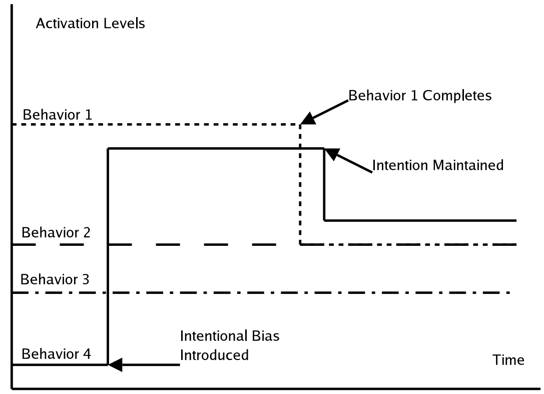


Fig. 9. An idealized graph of activation values for four behaviors demonstrating how intentional bias does not necessarily override existing behavioral expression. In this case the bias is not great enough to make behavior 4 run. The bias remains, however, and when behavior 1 completes, the bias provided to behavior 4 now allows it to run.

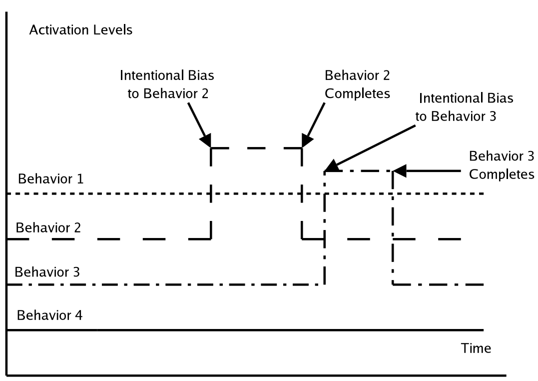


Fig. 10. An idealized graph of activation levels for a series of two behaviors before learned serialization. Behavior 2 is biased causing it to become active, followed by behavior 3.

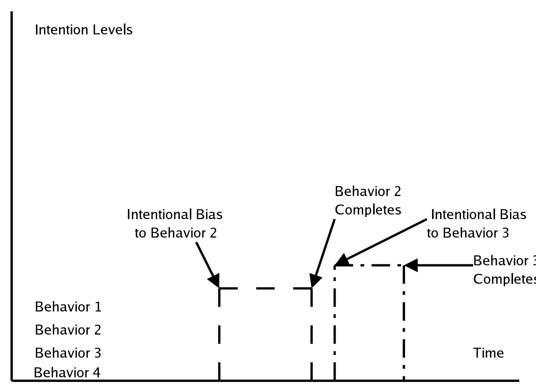


Fig. 11. An idealized graph of deliberately applied intentional values for a series of two behaviors before learned serialization.

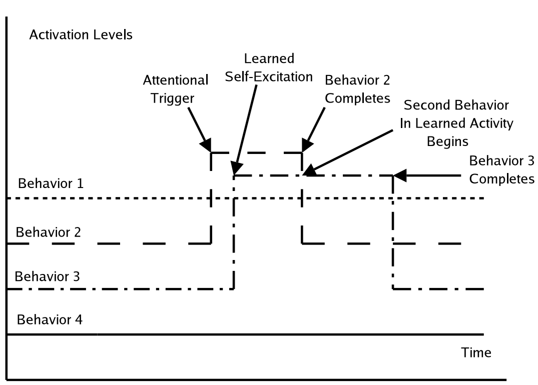


Fig. 12. An idealized graph of activation values for a learned routine behavior. In this example, after behavior 2 becomes active, behavior 3's activation increases via self-excitation in anticipation of being active next.

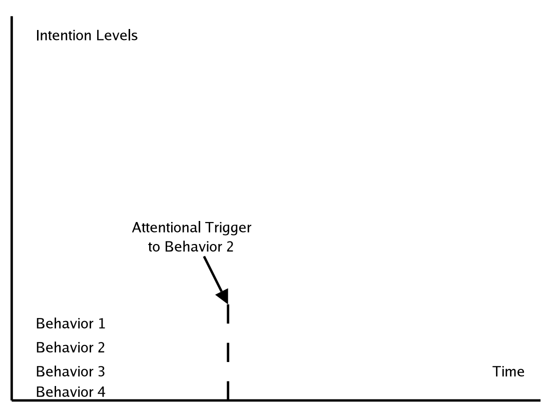


Fig. 13. An idealized graph of intentional activity showing an attentional trigger for a learned routine behavior. The trigger causes behavior 2 to become active and allows the complete learned routine to express itself automatically.

layer can relay a message to the deliberative system indicating that a task has become routine. Upon receipt of this message, the deliberative system shifts from providing a continuous intentional influence to instead providing a short attentional signal to start the sequence of actions corresponding to that plan. This occurs through the intentional bus using the same mechanism that handles intentional signals. Once this signal from the low-level schema has been received, the current sequence of actions can be marked as routine at the deliberative level. If the series of behaviors has in fact been learned, in the future the deliberative system can send a request to the intentional bus that an attentional message be sent to the first behavior in the sequence whenever that sequence is required to be re-activated. This attentional trigger is sent using the same mechanism that intentional bias is delivered to the targeted schema, with the only difference being the duration of the bias. For an attentional trigger, the intentional bus only sends the bias once. After that, the behavior is triggered and the bias is set back to zero instead of being actively maintained at a specific level. The first several times the attentional signal is sent, however, it is desirable for the deliberative system to monitor the execution of the sequence to ensure the sequence has indeed been learned properly. In the case that it has, the deliberative system no longer interacts with the subsequent schema in the sequence but instead allows the sequence to execute automatically at the reactive level. An illustration of the intentional and attentional activity before and after a serial action is learned can be seen in figures 10 to 13. Figure 10 shows the activation levels of four behaviors during the execution of a 2-step serial action. In this case, behavior 2 is biased through the intentional bus followed by behavior 3. The intentional activity during this process appears in figure 11. Figures 12 and 13 show the interplay between the attentional triggering and the execution of the learned serial behavior. Here, the attentional trigger replaces the constant intentional bias and the behavior is executed automatically at the reactive layer.

V. Design and Implementation of an Architecture for Deliberative Control via Intentional Bias

On overview of the deliberative/reactive architecture utilizing intentional bias can be seen in figure 14. The deliberative system can be viewed as a collection of interacting subsystems. These subsystems provide mechanisms for planning, plan execution, state tracking, as well as the intentional bus for interfacing the deliberative layer with the reactive layer.

A. Planner, Knowledge base, and APR Converter

Generation of plans is one of the primary functions of a deliberative system. To accomplish this task a suite of three components is utilized: the planner, the knowledge base, and the abstract plan representation (APR) converter. The knowledge base contains facts and assertions about the world, the state of the robot, the tasks that must be accomplished, as well as knowledge concerning the different behaviors the robot can execute and their effect upon the world. The knowledge base also contains the information required for the generation of the bias values in a plan. This knowledge takes the form of scalar values representing user and robot preferences for tasks. Future work will look into means of acquiring these values over the lifetime of the robot using on-line learning algorithms. The planner uses the information contained in the knowledge base to generate plans based on the goals of the robot at a given time.

To generate plans for execution in this architecture, a hierarchical task network planner was used for the initial implementation, which outputs plans consisting of ordered sequences of actions that need to be accomplished in order to achieve the goals maintained in the knowledge base. This Abstract Plan Representation (APR) can vary based on the specific type of planner being used. We do not use the planner's output directly so as not to tie the deliberative system to any one particular planner implementation for flexibility. As a result, other planners may be integrated without loss of generality. The APR converter serves as the glue between the planner and rest of the deliberative system by taking the planner's APR output and converting it into the internal plan representation (IPR). The resulting IPR is utilized by the IPR executor module on-board the robot to realize the generated plans.

B. Internal Plan Representation

As the APR output of the planner can vary based on the type of planner used, a formal language was developed as an internal plan representation (IPR) for the deliberative system. It serves as a concrete target for the plan format that the robot will execute. The IPR is composed of a series of statements detailing the steps generated by the planner. Each step is composed of a number of components, the first being a state requirement, detailing the conditions that must be true in the world for that particular part of the plan to be applicable. The current possible state qualifiers include the presence of a stimulus in short-term memory, the posture of the robot, time constraints, energy constraints, and plan progress. Additional qualifiers can be added readily as the requirements for state description expand along with the types of tasks needed.

The second major component is the schema that is to be biased during that step in the plan. This is followed by any binding parameters they may be applicable for the schema. For example, a reaching schema may be bound to the particular object that the robot may reach for, such as a ball. The binding parameter is followed by a priority value indicating the relative importance of the schema. This is used both to resolve resource conflicts in the execution of concurrent plans as well as serving as the magnitude of the intentional bias sent to that schema.

The final component is a flag to determine if the schema in that plan item is part of a learned action sequence and hence requires an attentional trigger rather then intentional bias.

More formally an IPR statement takes the form of:

$$s \;\; a(b) \;\; m \;\; i; \tag{4}$$

where

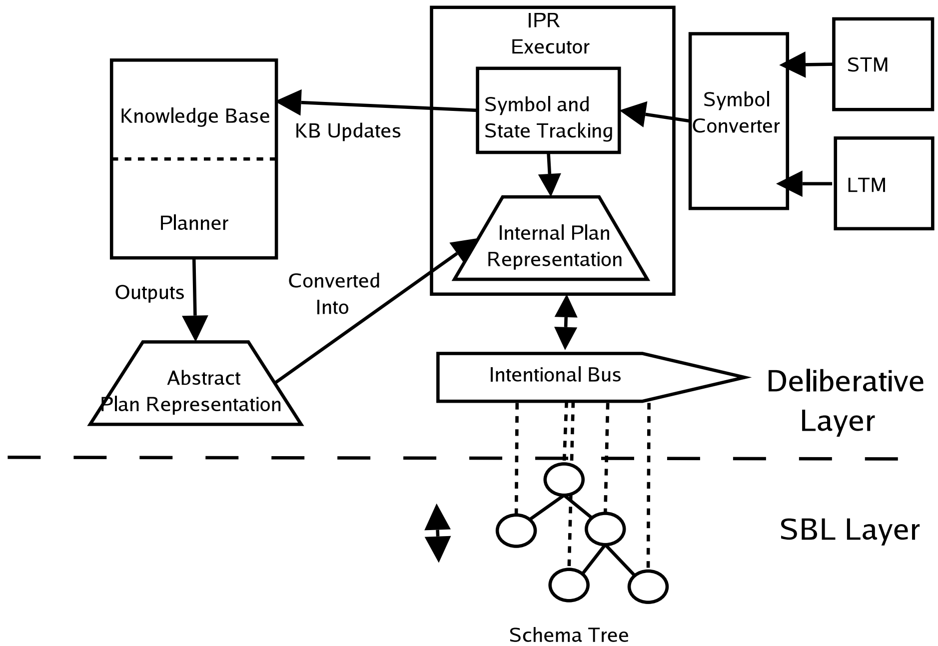


Fig. 14. Overview of the end-to-end deliberative/reactive architecture

TABLE I
SAMPLE IPR FOR A PLAN GENERATED FOR ATTENDING SOCCER CLASS

```
NOT Completed(1) SEARCH(LANDMARK3(#1) 75 false;
// If the first plan step has not been  finished then bind instance 1 of landmark 3 to the search
//    behavior search and bias the search behavior  with magnitude 75.

Completed(1) AND Present(LANDMARK3) APPROACH(LANDMARK3#1) 75 false;
// If the first plan step has finished and landmark 3 is present & then bind instance 1 of landmark
//    3 to the approach behavior and bias the approach behavior with magnitude 75.

Completed(2) SEARCH(PINKBALL#1) 75 (1,2,3) true;
// if the second plan item has been completed bind the search behavior to instance 1 of the pink
//    ball and send an attentional trigger of magnitude 75 (this will cause the routine behavior for
//    soccer to begin).
```

- s is the state over which the IPR item is valid where s takes the form of a statement of conjuncts and disjuncts of state variables s_1 to s_n.
- a is the schema that will be intentioned.
- b is the binding parameter for the schema where b is a couple, $< object, identifier >$, representing the instance of a particular object to bind to this schema
- m is an integer parameter indicating the magnitude of the desired bias
- i is a boolean flag indicating whether the schema should be maintained by the intentional bus or be serviced by an attentional trigger.

Table I shows a small sample of a plan in the IPR format. A single IPR statement is called a plan item as it corresponds to the smallest unit of a particular plan. Once the abstract plan has been converted into the IPR, it is loaded into the plan working memory of the IPR executor.

C. *Symbol Conversion Module*

The symbol conversion module is largely responsible for translating stimuli, both from the outside world and internally, into a form that is usable by the deliberative system. Internal stimuli include internal state variables such as energy levels while external stimuli can include things such as visual or audio input. Using updates from the internal state module in the EGO architecture, the symbol conversion module copies the relevant data into the current world state representation used by the IPR executor. In addition, the symbol conversion module examines the contents of short-term memory for various external stimuli. If it finds any stimuli in short term memory, it determines what they are (faces, landmarks, etc) as well as their associated target ID. It translates any stimuli it finds into a name representing the stimuli and places it into the symbol buffer held in the IPR executor for potential binding as well as for state qualification evaluation for the plan items.

D. IPR Executor

The IPR executor is the core module for the execution of plans generated by the deliberative system. Its functions include: plan item selection, plan progress tracking, symbol binding, and schema conflict resolution. The primary task for the IPR executor is the determination of applicable plan items based upon their state requirements and the current state of the world. This process occurs in several stages. The first phase selects the set of plan items which are not executing and whose state requirements currently match those of the world state. This set is then tested for potential schema conflicts (described below). The plan items without conflicts are then sent to the symbol binding mechanism. After binding they are then set to *running*, and a message is sent to the intentional bus indicating which schema to bias and the associated bias magnitude as specified in the IPR statement. In addition, the message contains the appropriate flag designated by the plan item indicating if it is an intentional request requiring maintenance by the bus of the behavioral bias or if it is instead an intermittent attentional trigger used for a routine activity.

The second major task the IPR executor performs is the binding of symbols to the schema in plan items. This produces a result similar to a mechanism currently found in the EGO architecture that allows children schema of the same parent to share target information. This rebinding allows different plan items to share the same target information despite belonging to different subtrees in the underlying behavioral tree. The IPR executor binds the objects in the symbol buffer to those needed by the running schemas based on the binding parameter in their plan item. All schemas that have the same binding parameter are informed of the target ID.

Schema conflict resolution, the third function of the IPR executor, refers to the fact that multiple plan items may be in competition for a single schema. For example, there may be two potentially active plan items which both require the use of an approach schema for two different objects. Only one of these plan items can be active at a particular time. The IPR executor uses a priority-based arbitration mechanism to resolve these conflicts where the priority for a particular plan item is represented by the magnitude of the intentional bias as specified in the IPR representation. Whenever a plan item is applicable, the IPR executor will first check that the appropriate schema is not already in use by another plan item. If it is, it compares their bias magnitudes and selects the plan item with the highest magnitude. In the case of a tie, the plan item that currently has control of the schema remains the executing schema.

Related to schema conflict resolution is the notion of parallel IPR plan item execution. Barring symbol conflicts such as described above, schema conflicts due hardware resource conflict (i.e. two competing schema that each cause the right arm to move to serve its own ends) are handled by the underlying action selection mechanism. Within the EGO architecture, multiple behaviors can be active as long as they do not result in hardware resource conflicts. As the intentional system described herein only biases behavior activation levels and does not activate the behaviors directly, IPR plans inherit the parallelizability of the existing underlying action selection mechanism. This allows multiple IPR plan statements to safely execute as long as there are no underlying hardware resource conflicts.

The final task of the IPR executor is tracking plan progression, notably, which plan items are running and which plan items have been completed. To do this, the IPR executor queries the intentional bus to determine if a particular schema belonging to a plan item has completed. If it has, the IPR executor sends a request to the intentional bus to set the intention value to zero. It then unbinds the symbols for that plan item and updates the requisite deliberative state information to indicate that the plan item has been completed.

E. Intentional Bus

As described previously, the intentional bus serves as the interface between the deliberative system and the underlying reactive layer. In this implementation, it serves as the interface between the IPR executor and the SBL. The intentional bus stores a large amount of information about the underlying schemas such as their state (ready, active, sleeping, etc.), their current activation levels, their process ids, and their current intentional levels. Messages sent from the SBL concerning schema status and activation levels are routed to the intentional bus for determining schema completion. The primary function of the intentional bus is to influence the underlying SBL layer via intentional threads which serve as biases, both excitatory and inhibitory, in the activation of schema.

Upon receipt of a message from the IPR executor indicating a specific schema to bias, the intentional bus stores the requested bias magnitude for later bias calculation as well as the flag indicating if it corresponds to an attentional trigger or intentional bias. In the case that the magnitude is zero, the intentional bias is immediately stopped for that schema. At every timestep, the intentional bus examines the bias magnitude values for each schema, and uses this value to calculate the appropriate intentional bias to send to the associated schema. The function used for bias calculation in this implementation is

$$\frac{m \sum a_i}{100}, \tag{5}$$

once again, where a_i is the activation value for schema i. Thus for priorities less then 100, bias can be introduced without preventing other schema from running. For values greater than 100, the plan item is certain to execute. Values less than 0 result in an inhibitory effect. For any bias value that is non-zero, the intentional bus sends a message to the SBL indicating the bias value that should be exerted on the behavior. If the intentional bus has been requested to send an attentional trigger to a particular schema, the next timestep the intentional level is set to zero.

VI. Evaluation

To demonstrate the ability of the implemented system to provide both goal-oriented behavior concurrent with pre-existing action selection mechanisms, two experiments were

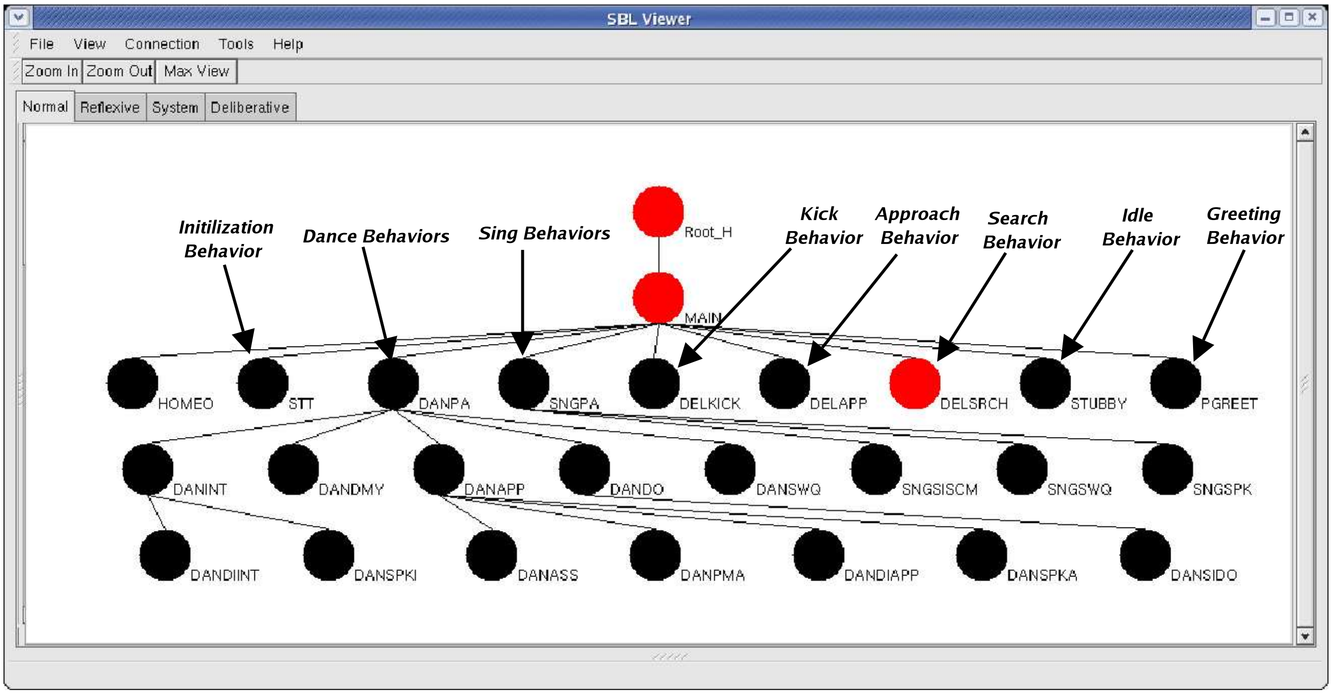


Fig. 15. Behavioral tree used in intentional experiments. The bold-italic labels indicate the behaviors associated with that node and any of the nodes children in applicable.

conducted. In the first experiment we look purely at the plan execution capabilities of the system by the addition of large amounts of intentional bias ($m > 100$). A behavioral tree was used consisting of several behaviors including dancing, singing, approaching, searching, kicking, a reflex action to turn towards loud noise, and a greeting action (figure 15). These behaviors were a combination of pre-existing behaviors developed by Sony staff as well as custom designed behaviors.

In the first experiment, a plan is generated to play soccer consisting of the three steps shown in Table II, namely searching for, approaching, and then kicking a pink ball. While this plan is being executed by QRIO, the operator calls out to QRIO to try to get its attention to interact. The operator then shows QRIO a green ball, the releasing mechanism for the greet behavior. The greet behavior, if activated, causes QRIO to stop, wave, and then sit down for a few moments.

The activation level traces for each behavior appear in figure 16 while the intentional levels are shown in figure 17. Figure 18 depicts the experiment while it is being executed. As seen from the activation levels, each item of the plan is executed in sequential order mimicking the effect of a traditional deliberative controller directing the behavior of the robot. Note the large spike in activation level for the approach behavior D (figure 16) while QRIO is executing the plan. This occurs when the operator shouted and tried to gain QRIO's attention and showed QRIO the green ball. The greeting behavior's activation level increased greatly but the active maintenance of the intentional levels by the deliberative system prevents the high priority plan from being interrupted as shown by C in figure 17. In the second experiment, we used a slightly modified IPR for the soccer plan, this time reducing the priority of each behavior in the plan sequence as shown in Table III. The plan is then executed as before, with the operator attempting to distract QRIO. Graphs of the activation and intentional levels can be seen in figures 19 and 20 respectively while figure 18 shows the experiment. This time, when the operator tries to get QRIO's attention and shows him the green ball, the activation level exceeds that of the particular plan item, even with active maintenance (Figure 19, D). QRIO interrupts the plan's execution to interact with the user before continuing on later with the original plan. This example demonstrates how the deliberative system allows for the augmentation of the robot's normal behavior without interfering with its underlying action selection mechanism.

VII. Conclusions and Future Work

This work presents the concept of intentional bias for deliberative control. Through behavioral biasing we show that goal-oriented behavior can be exhibited without overriding the present underlying action selection mechanisms. In addition, through the active modulation of bias afforded by the intentional bus, the deliberative system is provided with a simple and uniform mechanism for interfacing with a reactive controller. In our implementation, we have investigated an additive mechanism for incorporating intentional bias into the reactive layer and demonstrated this experimentally. Further research is necessary to determine if non-additive methods may be more effective in modifying the activation levels of the behaviors. In addition, it may be of value to investigate alternative means of calculating the magnitude of the bias for systems that do not have uniform activation level calculations over all behaviors.

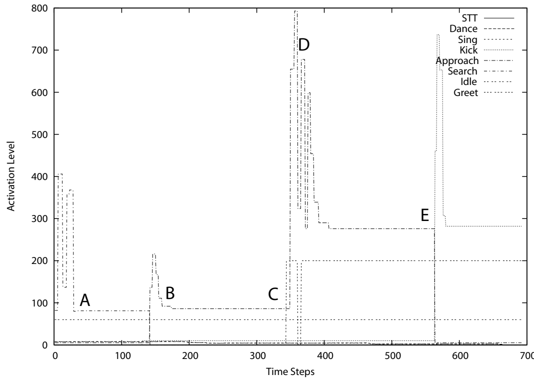


Fig. 16. Activation levels trace during the execution of a high intentional magnitude plan: A) Search behavior is active. B) Approach behavior becomes active. C) Releasing mechanism for greet behavior is found and its activation is increased. D) Intentional bus maintains appropriate intention levels keeping approach active. E) Kick behavior becomes active.

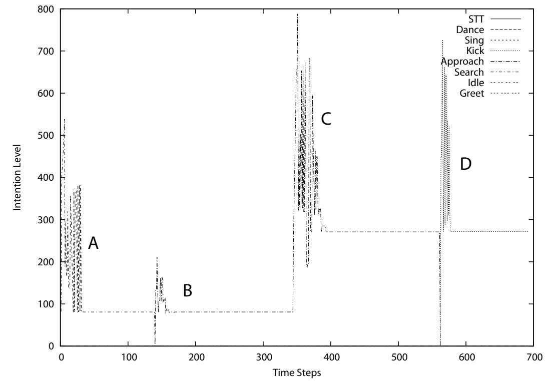
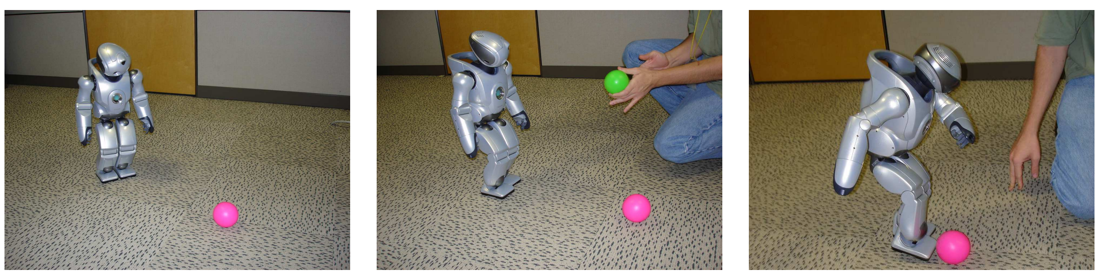


Fig. 17. Intention levels trace during the execution of a high intentional magnitude plan: A) Intentional bias is sent to search behavior. B) Intentional bias is stopped for search and started for approach behavior. C) Intentional bias is increased by intentional bus due to changes in schema activation levels. D) Intentional bias to approach is stopped and bias to kick behavior begins.



(a) **(b)** **(c)**

Fig. 18. Depiction of high intention soccer plan. a) QRIO approaches the ball b) The operator tries to distract QRIO with the green ball (the releasing mechanism for the greet behavior) c) QRIO kicks the ball and completes the plan.

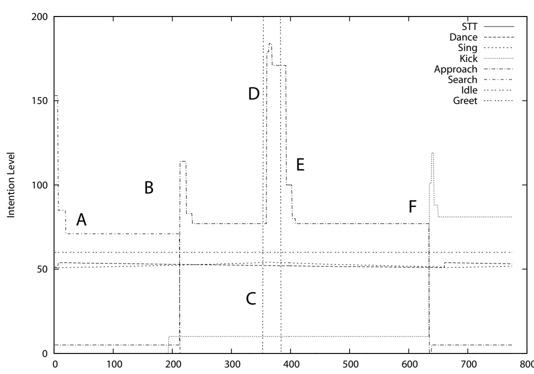


Fig. 19. Activation levels trace during the execution of a low intentional magnitude plan: A) Search behavior is active. B) Approach behavior becomes active. C) Releasing mechanism for greet behavior is found and its activation is increased. D) Intentional bus maintains appropriate intention levels but intentional magnitude is low enough to allow other behaviors to execute. E) Greeting behavior finishes and approach behavior becomes active again. F) Kick behavior becomes active.

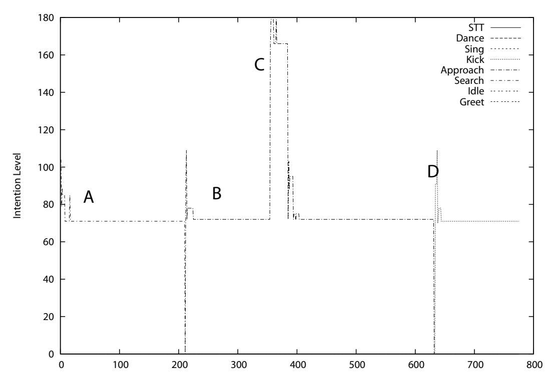


Fig. 20. Intention levels trace during the execution of a low intentional magnitude plan: A) Intentional bias is sent to search behavior. B) Intentional bias is stopped for search and started for approach behavior. C) Intentional bias is increased by intentional bus due to changes in schema activation levels. D) Intentional bias to approach is stopped and bias to kick behavior begins.

TABLE II

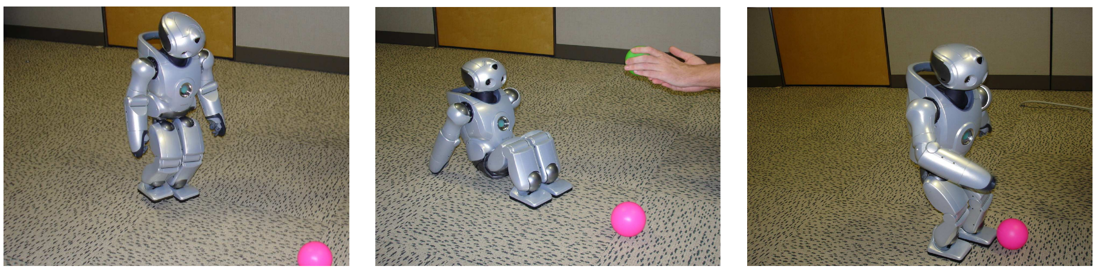
IPR REPRESENTATION FOR THE HIGH INTENTIONAL SOCCER PLAN

```
// High intentional bias plan
NOT Present(PinkBall) SEARCH(PINKBALL#1) 101;
Completed(1) AND Present(PINKBALL) APPROACH(PINKBALL#1) 101;
Completed(2) KICK(PINKBALL#1) 101;
```

TABLE III

IPR REPRESENTATION FOR THE LOW INTENTIONAL SOCCER PLAN

```
// Low intentional bias plan
NOT Present(PinkBall) SEARCH(PINKBALL#1) 75;
Completed(1) AND Present(PINKBALL) APPROACH(PINKBALL#1) 75;
Completed(2) KICK(PINKBALL#1) 75;
```



(a) **(b)** **(c)**

Fig. 21. Depiction of low intention soccer plan. a) QRIO approaches the ball b) The operator tries to distract QRIO with the green ball, the lower intentional bias of this plan allows the greeting behavior to run and QRIO greets the user and sits down. c) QRIO gets back up, continues executing the plan, and kicks the ball.

ACKNOWLEDGMENT

We would like to thank Ken-ichi Hidai, Kuniaki Noda, as well as the rest of the Sony Intelligence Dynamics Laboratories staff for their support in this work.

REFERENCES

[1] W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, V. Steiner, and S. Thrun, "Interactive museum tour-guide robot," in *Proceedings of AAAI-98*, 1998, pp. 11–18.

[2] C. Brazeal, "Sociable machines: Expresive social exchange between humans and robots," Ph.D. dissertation, MIT, 2000.

[3] L. Moshinka and R. Arkin, "On TAMEing robots," in *Proceedings IEEE Conference on Systems, Man, and Cybernetics*, 2003.

[4] D. A. Norman and T. Shallice, *Consciousness and Self Regulation: Advances in Theory and Research*. Academic Press, 1986, vol. 4, ch. Attention to action: Willed and automatic control of behavior, pp. 515–549.

[5] F. Mastrogiovanni, A. Sgorbissa, and R. Zaccari, "A system for hierarchical planning in service mobile robots," in *Proceedings of IAS08*, 2004.

[6] K. Okada, A. Haneda, H. Nakai, M. Inaba, and H. Inoue, "Environment manipulation planner for humanoid robots using task graph that generates action sequence," in *Proceedings of 2004 International Conference on Intelligent Robots and Systems*, 2004.

[7] R. Arkin, *Behavior-Based Robotics*. MIT Press, 1998.

[8] E. Gat, *Artificial Intelligence and Mobile Robots*. MIT/AAAI Press, 1997, ch. On three-layer architectures.

[9] D. Lyons and A. Hendricks, "Planning for reactive robot behavior," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1992, pp. 2675–2680.

[10] D. Payton, J. Rosenblatt, and D. Keirsey, "Plan guided reaction," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 6, pp. 1370–1382, 1990.

[11] A. Wagner and R. Arkin, "Internalized plans for communication-sensitive team behaviors," in *Proceedings of the International Conference on Intelligent Robotics and Systems*, 2003, pp. 2480–2487.

[12] E. Beaudry, Y. Brosseau, C. Cote, C. Raievsky, D. Letourneau, F. Kabanza, and F. Michaud, "Reactive planning in a motivated behavioral architecture," in *Proceedings of the Reactive Planning in a Motivated Behavioral Architecture*, 2005, pp. 1242–1247.

[13] J. Rosenblatt, "DAMN: A distributed architecture for mobile navigation," in *Proceedings of the AAAI Spring Symposium on Lessons Learned from Implemented Software Architectures for Physical Agents*, 1997.

[14] P. Carpenter, P. Riley, M. Veloso, and G. Kaminka.

[15] R. Cooper and T. Shallice, "Modeling the selection of routine action: Exploring the criticality of parameter values," in *Proceedings of the 19th Annual Conference of the Cognitive Science Society*, 1997, pp. 130–135.

[16] ——, "Contention scheduling and the control of routine activities," *Cognitive Neuropsychology*, vol. 17, no. 4, pp. 297–338, 2000.

[17] V. Andronache and M. Scheutz, "Contention scheduling: A viable action-selection mechanism of robotics?" in *Proceedings of the 13th Midwest AI and Cognitive Science Conference*. AAAI Press, 2002, pp. 122–129.

[18] D. Glasspool, "The integration of control and behavior: Insights from neuroscience and AI," in *Proceedings of the How to Design a Functioning Mind Symposium at AISB-2000*, 2000.

[19] T. Schallice and P. Burgess, "The domain of supervisory processes and temporal organization of behavior," in *Philosophical Transactions of the Royal Society of London B*, vol. 351, 1996, pp. 1405–1412.

[20] J. Garforth and A. Meehan, "Driven by novelty? Integrating executive attention and emotion for autonomous cognitive development," in *Proceedings of the Developmental Robotics AAAI Spring Symposium*, 2005.

[21] J. Garforth, A. Meehan, and S. McHale, "Attentional behavior in robotic agents," in *Proceedings of the International Workshop on Recent Advances in Mobile Robots*, 2000.

[22] B. Blumberg, "Action-selection in hamsterdam: Lessons in ethology," in *Proceedings of the Third International Conference on the Simulation of Adaptive Behavior*, 1994, pp. 108–117.

[23] P. Maes, "How to do the right thing," *Connection Science Journal*, vol. 1, no. 3, pp. 291–323, 1989.

[24] D. Terzoupoulos, X. Tu, and R. Grzeszczuk, "Artificial fishes: Autonomous locomotion, perception, behavior, and learning in a simulated physical world," *Artificial Life*, vol. 1, no. 4, pp. 327–351, 1994.

[25] T. Tyrrell, "Computational mechanisms for action selection," Ph.D. dissertation, University of Edinburgh, 1993.

[26] M. Fujita, Y. Kuroki, T. Ishida, and T. Doi, "Autonomous behavior control architecture of entertainment humanoid robot SDR-4X," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003, pp. 960–967.

[27] R. Arkin, M. Fujita, T. Takagi, and R. Hasegawa, "An ethological basis for human-robot interaction," *Robotics and Autonomous Systems*, vol. 42, no. 3-4, 2003.

[28] Y. Hoshino, T. Takagi, U. D. Profio, and M. Fujita, "Behavior description and control using behavior module for personal robot," in *Proceedings of the 2004 International Conference on Robotics and Automation*, 2004, pp. 4165–4171.

[29] R. Bonasso and D. Kortenkamp, "An intelligent agent architecture in which to pursue robot learning," in *Working Notes: MCL-COLT '94 Robot Learning Workshop*, 1994.

[30] S. Cheranova and R. Arkin, "From deliberative to routine behaviors: A cognitively-inspired action selection mechanism for routine behavior capture," in review.