# TWO LAYERS ACTION INTEGRATION FOR HRI
## *Action Integration with Attention Focusing for Interactive Robots*

Yasser Mohammad and Toyoaki Nishida
*Graduate School of Informatics, Kyoto University, Kyoto, Japan*

Keywords: HRI, Action Selection, Natural Listening.

Abstract: Behavior architectures are widely used to program interactive robots. In these architectures multiple *behaviors* are usually running concurrently so a mechanism for integrating the resulting actuation commands from these *behaviors* into actual actuation commands sent to the robot's motor system must be faced. Different architectures proposed different action integration mechanisms that range from distributed to central integration. In this paper an analysis of the special requirements that HRI imposes on the action integration system is given. Based on this analysis a novelle hybrid action integration mechanism that combines distributed attention focusing with a fast central integration algorithm is presented in the framework of the EICA architecture. The proposed system was tested in a simulation of a listener robot that aimed to achieve human-like nonverbal listening behavior in real world interactions. The analysis of the system showed that the proposed mechanism can generate coherent human-like behavior while being robust against signal correlated noise.

## 1 INTRODUCTION

Behavioral architectures are widely used to program interactive robots (Ishiguro et al., 1999), (S Karim, 2006). Behavioral architectures employs a vertical decomposition of the software design into a set of co-running *behaviors* that continuously map the current state of the environment into commands to the actuators of the robot. A basic problem that must be solved in any such architecture is how to integrate the results of multiple running processes into a final command sent to the actuators of the robot. The proposed solution to this problem (referred to as the action integration problem hereafter) can be divided into selective solutions that selects the action of a single behavior at any point of time to control the robot and combinative solutions that generates the final commands based on the proposed responses of multiple behaviors and hybrid solutions that tries to combine cooperation with coordination. Another dimension of comparison between action integration solutions is whether or not a central integrator is employed. According to this dimension available solutions are central or distributed. The action integration mechanism proposed in this paper can be classified as a hybrid two layered architecture with distributive attention focusing behavior level selection followed by a fast central combinative integrator.

The next section formalizes the requirements for an action integration mechanism suitable for HRI applications. Section 3 introduces the $L_0$EICA architecture for which this action integration mechanism was designed. Section 4 gives the details of the action integration mechanism proposed in this paper and section 5 reports an experiment with a humanoid robot that used the proposed action integration mechanism.The paper is then concluded.

## 2 REQUIREMENTS

One goal of HRI research is to realize robots that can use human-like verbal and nonverbal interaction modalities. To achieve this goal researchers usually employ behavioral architectures because of their robustness and fast response. A special property of HRI applications is that the details of the external behavior of the robot will be interpreted by human users as intentional signals (Breazeal, 2005), and usually combining nonverbal messages results on absolutely different messages. This is why combinative architectures are not common in HRI research and most of the HRI specific architectures employ a selective



Mohammad Y. and Nishida T. (2008).
TWO LAYERS ACTION INTEGRATION FOR HRI - Action Integration with Attention Focusing for Interactive Robots.
In *Proceedings of the Fifth International Conference on Informatics in Control, Automation and Robotics - RA*, pages 41-48
DOI: 10.5220/0001482600410048

integration strategy. For example (Ishiguro et al., 1999) proposed a robotic architecture based on situated modules and reactive modules in which reactive modules represent the purely reactive part of the system, and situated modules are higher levels modules programmed in a high-level language to provide specific behaviors to the robot. The situated modules are evaluated serially in an order controlled by the module controller. Research in nonverbal communication in humans reveals a different picture in which multiple different processes do collaborate to realize the natural action. For example (Argyle, 2001) showed that human spatial behavior in close encounters can be modeled with two interacting processes. It is possible in the selective framework to implement these two processes as a single behavior but this goes against the spirit of behavioral architectures that emphasizes modularity of behavior (Perez, 2003). This leads to the first requirement for HRI aware action integration: *The action integration mechanism should allow a continuous range from pure selective to pure combinative strategies*. In other words the system should use a *hybrid* integration strategy. The need to manage the degree of *combinativity* based on the current situation entails the second requirement: *The action integration mechanism should adapt to the environmental state using timely sensor information as well as the internal state of the robot*. In current systems this requirement is usually implemented by using a higher level deliberative layer but in many cases the interaction between simple reactive within the action integrator can achieve the same result as will be shown in the example implementation of this paper.

The Hybrid Coordination approach presented in (Perez, 2003) is the nearest approach to achieve this first requirement. In this system every two behaviors are combined using a *Hierarchical Hybrid Coordination Node* that has two inputs. The output of the HHCN is calculated as a nonlinear combination of its two inputs controlled by the activation levels of the source behaviors and an integer parameter $k$ that determines how combinative the HHCN is, where larger values of $k$ makes the node more selective. The HHCNs are then arranged in a hierarchical structure to generate the final command for every DoF of the robot (Perez, 2003). Although experiments with the navigation of an autonomous underwater robot have shown that the hybrid coordination architecture can outperform traditional combinative and selective architectures, it still has some limitations in the HRI domain. One major limitation of the hybrid coordination system is its reliance on binary HHCNs which makes it unsuitable for large numbers of behaviors due to the exponential growth in the number of HHCNs needed. Another problem is the choice of the parameter $k$ for every HHCN. Yet the most difficult problem for this system is figuring out the correct arrangement of the behaviors into the HHCN inputs. This leads to the third requirement: *The action integration mechanism should not depend on global relationships between behaviors*. One of the major problems with this architecture is that every behavior must calculate its own activation level. Although this is easy for behaviors like *avoid-obstacles* or *go-to*, it is very difficult for interactive processes like *attend-to-human* because the achievement of such interactive processes is not manifested in an easily measurable specific goal state that must be achieved or maintained but in the exact way the overall behavior of the robot is changing over time. This leads to the fourth requirement: *The action integration mechanism should separate the calculation of behavior's influence from the behavior computation*.

The number of behaviors needed in interactive robots usually is very high compared with autonomously navigating robots if the complexity of each behavior is kept acceptably low, but most of those behaviors are usually passive in any specific moment based on the interaction situation. This property leads to the fifth requirement: *The system should have a built-in attention focusing mechanism*. HRI systems usually work in the real world with high levels of noise but it is required that the robot shows a form of goal directed behavior. This leads to the sixth requirement: *The action integration system should be robust against noise and data loss to provide a goal-directed behavior*.

In summary the six requirements HRI imposes on the action integration system are:

R1 It should allow a continuous range from pure selective to pure combinative strategies

R2 It should adapt to the environmental state utilizing timely sensor information.

R3 It should not depend on global relationships between behaviors

R4 It should separate the calculation of behavior's influence from the behavior's computation

R5 It should have built-in attention focusing

R6 It should be robust against noise and data loss.

Table 2 compares the action integration scheme of some well used behavioral architectures with the proposed system in terms of the six requirements

In this paper an action integration mechanism that has the potential of meeting these requirements is presented.

Table 1: Comparison of the Action Integration Capabilities of Some Behavioral Architectures in Terms of the Six Requirements in section 2.

| Architecture | Integration | Implement. | R1 | R2 | R3 | R4 | R5 | R6 |
|---|---|---|---|---|---|---|---|---|
| Subsumption (Brooks, 1986) | Selective | Distributed | | | √ | | √ | √ |
| ASD (Maes, 1989) | Selective | Distributed | | | | √ | | √ |
| PDL (Steels, 1993) | Combinative | Central | | | √ | | | |
| Motor Schemas (Arkin, 1993) | Combinative | Distributed | | | √ | | √ | |
| Hybrid Coordination (Perez, 2003) | Hybrid | Distributed | √ | | | | | √ |
| AVB (Nicolescu and Matarić, 2002) | Selective | Distributed | | √ | √ | | | √ |
| Situated Modules (Ishiguro et al., 1999) | Selective | Central | | √ | √ | | √ | √ |
| Proposed System | Hybrid | Hybrid | √ | √ | √ | √ | √ | √ |

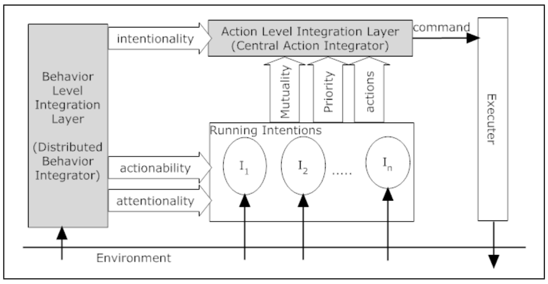


Figure 1: The Proposed Action Integration Mechanism.

## 3 $L_0$EICA

EICA is a behavioral hybrid architecture designed for HRI applications (Mohammad and Nishida, 2007). The basic behavioral element of EICA is the *intention*. Every intention implements a simple well defined reactive capability of the robot. Intentions are more like motor schema of (Arkin, 1993) than complete behaviors. Every intention has three attributes:

**Attentionality** a real number (0 1) that specifies the relative speed at which the intention is running.

**Actionability** a real number (0 ∞) specifying the activation level of the intention. A zero or negative activation level prevents the intention from execution. This attribute controls the influence of this intention on other running components.

**Intentionality** a real number (0 ∞) set by the distributed action integration layer and used by the central integrator to combine the actions of various intentions.

## 4 THE PROPOSED SYSTEM

Fig. 1 shows the block diagram of this hybrid system. The following subsections will detail the two layers of the system.

### 4.1 Behavior Level Integration

The goal of this layer of the action integration mechanism is to provide timely values for the *actionability*, *attentionality*, and *intentionality* for various intentions in the system. The first two of those parameters will determine how frequent every intention will be allowed to send actions to the central action integrator, and the third will be used by the action integrator, along with other parameters, to integrate the actions proposed by all the running intentions as will be detailed in the next section.

This layer is implemented as a set of *processes* connected together and to the intentions through *effect channels*.

Every process runs with a speed directly proportional to its *attentionality* level as long as its *actionability* is positive. The *processes* are connected together through *effect channels* in a network. Every *effect channel* has a set of $n$ inputs that use continuous signaling and a single output that is continuously calculated from those inputs according to the *operation* attribute of the *effect channel*. The example presented in this paper uses the *Avg* operation defined as: $y = \frac{\sum_{i=1}^{n}(a_i x_i)}{\sum_{i=1}^{n}(a_i)}$, where $x_i$ is an input port, $a_i$ is the *actionability* of the object connected to port $i$ and $y$ is the output of the effect channel. Every *process* or *intention* has a single effect channel connected to its *attentionality* attribute and another effect channel connected to its *actionability* attribute, and effect channels can be arranged into hierarchical structures like the HHCNs in (Perez, 2003) but rather than carrying action and influence information, the effect channels carry only the influence information between different processes of the system.

## 4.2 Action Level Integration

This layer consists of a central fast action integrator that fuses the actions generated from various intentions based on the intentionality of their sources, the mutuality and priority assigned to them, and general parameters of the robot. The *actionability* of the action integrator is set to a fixed positive value to ensure that it is always running. The action integrator is an *active* entity and this means that its *attentionality* is changeable at run time to adjust the responsiveness of the robot. The action integrator periodically checks its internal master *command* object and whenever finds some action stored in it, the *executer* is invoked to execute this action on the physical actuators of the robot.

Algorithm 1 shows the *register-action* function responsible of updating the internal *command* object based on actions sent by various intentions of the robot. The algorithm first ignores any actions generated from intentions below a specific system wide threshold $\tau_{act}$. The function then calculates the *total priority* of the action based on the intentionality of its source, and its source assigned priority. Based on the mutuality assigned to every degree of freedom (DoF) of the action, difference between the total priority of the proposed action and the currently assigned priority of the internal command, the system decides whether to combine the action with the internal command, override the stored command, or ignore the proposed action. Intentions can use the *immediate* attribute of the *action* object to force the action integrator to issue a command to the executer immediately after combining the current action.

## 4.3 Requirement Achievement

1. The system can achieve a continuous range of combinative to selective behavior based on the values of the actionability and attentionality of the intentions and the mutuality assigned to the actions. On one extreme $\tau_{act}$ can be set to the same value of the intention with the highest intention which will lead to a purely selective behavior. On the other hand $\tau_{act}$ and action mutuality can be set to zero while keeping the intentionality of all intentions equal which will lead to a purely combinative behavior.

2. The distributed layer of the system can use the sensor information and internal state stored in the processes to manipulate the $tau_{act}$ parameter as well as the intentionality of various intentions to control the level of combinative behavior based on timely information from the environment.

**Algorithm 1** : Register Action Algorithm.

**function** REGISTER-ACTION(Action $a$, Intention s)
- **if** $s.actionability < \tau_{act} \vee s.intentionality < \tau_{int}$ **then**
  - *exit*
- **end if**
- $c \leftarrow$ current combined command
- $p \leftarrow a.priority + max\_priority \times s.intentionality$
- **for** every DoF $i$ in the $a$ **do**
  - $combined \leftarrow true$
  - **if** $p < c.priority \wedge s \neq c.source$ **then**
    - $c.source \leftarrow s$
    - $c.dof(i) \leftarrow a.dof(i)$
    - $c.priority(i) \leftarrow p$
    - $c.hasAction(i) \leftarrow true$
  - **end if**
  - **if** $c.source \neq null$ **then**
    - $c.source \leftarrow null$
    - $c.priority(i) \leftarrow \max(p, c.priority(i))$
  - **else**
    - $c.source \leftarrow s$
    - $c.priority(i) \leftarrow p$
  - **end if**
  - **if** $a.mutual = true \vee c.source = s$ **then**
    - $c.source \leftarrow s$
    - $c.dof(i) \leftarrow a.dof(i)$
    - $c.priority(i) \leftarrow p$
  - **else**
    - $c.dof(i) \leftarrow \frac{p \times a.dof(i) + c.priority \times c.dof(i)}{p + c.priority}$
  - **end if**
  - **if** $combined = true$ **then**
    - *return false*
  - **end if**
  - $c.actionable \leftarrow true$
  - **if** $a.notCombinableWithLower$ **then**
    - $c.stopCombiningLower \leftarrow true$
  - **end if**
  - **if** $a.immediate$ **then**
    - *execute a*
  - **end if**
- **end for**

**end function**

3. The proposed action integration mechanism does not depend on any global relation between the intentions or the processes and adding new processes or intentions will only involve deciding the local effect, and data channels related to this new active component.

4. The influence of any intention on the final behavior of the robot is controlled based on its *intentionality* which is in turn controlled by the processes of the behavior integration layer rather than the intention itself. This provides the required separation between the calculation of influence and the computation of the basic behaviors.
5. Attention focusing is implemented in the lowest level of implementation in the proposed system and is controlled by the *attentionality* of various processes and intentions. This calculation is separated from the influence calculation through the *actionability* parameters. Most existing robotic architectures do not separate these two aspects of behavior control. By separating the actionability from the attentionality and allowing actionability to have a continuous range, EICA enables a form of attention focusing that is usually unavailable to behavioral systems. This separation allows the robot to select the active processes depending on the general context (by setting the actionability value) while still being able to assign the computation power according to the exact environmental and internal condition (by setting the attentionality value). The fact that the actionability is variable allows the system to use it to change the possible influence of various processes (through the operators of the effect channels) based on the current situation.
6. The central action integrator acts as a low pass filter that reduces the effects of noise by combining the actions sent by various intentions and accumulating them for a period that is controlled by the *attentionality* of the action integrator itself. This provides a simple means of noise rejection. A more subtle advantage of the proposed system in relation to goal-directed behavior is the distributive nature of the behavior integration layer which allows, for well designed robots, the emergence of complex external behavior from simple internal processes. The other option to achieve this complex external behavior was to map this complexity directly to the intentions themselves which would have complicated the design too much.

## 4.4 Limitations

Although the previous subsection has shown that the proposed architecture can theoretically achieve the six requirement of section 2 the system still has some limitations. One of the major problems with the proposed system is that it is sometimes difficult design the behavior level integration layer because of its asynchronous distributed nature and it is also difficult to learn the parameters needed to control the timing of the operation of various processes in it. Although careful division of the task into a set of intentions and task-specific integration processes can alleviate this problem, a general guideline to this process is needed to simplify the design of EICA robots. This limitation can be overcame by restricting the number of processes active at any moment to only one process but this will lead to over-complication in the design of the intentions. A better solution to this problem is a direction for future research.

# 5 EXPERIMENT

The ability to use human-like nonverbal listening behavior is an advantage for humanoid robots that coexist with humans in the same social space and is complex enough to test some of the proposed system's features. (Kanda et al., 2007) implemented a robot that tries to use natural human like body language while listening to a human giving it road directions based on the situated modules architecture. The road guidance task is simplified by the fact that there are no other objects of interest in the scene except the human. In this work we try to build a general listener robot that can generate natural nonverbal behavior in an explanation scenario involving unknown number of objects that can also be moving. As a minimal design, only the head of the robot was controlled during this experiment. This decision was based on the hypothesis accepted by many researchers in the nonverbal human interaction community that gaze direction is one of the most important nonverbal behaviors involved in realizing natural listening in human-human close encounters (Argyle, 2001). This example is intended as a guide for designing systems that can utilize the proposed integration strategy and a proof of its applicability for HRI.

## 5.1 Procedure

The evaluation data was collected as follows:

1. Six different explanation scenarios were collected in which a person is explaining the procedure of operating a hypothetical machine that involves pressing three different buttons, rotating a knob, and noticing results in an LCD screen in front of a Robovie II robot while pretending that the robot is listening to the explanation. The data was collected using the PhaseSpace Motion Digitizer system (PhaseSpace, 2007) by utilizing 18 LED markers attached to various parts of the speaker's body. The data was logged 460 times per second.

2. The logged data were used as the input to a robot simulator that implemented the proposed system. The behavior of the robot's head was compared with known human-human behavior in terms of mutual gaze and gaze toward instructor.

3. For every scenario 20 new synthetic scenarios were generated by utilizing 20 different levels of noise. The error level is defined as the percentage of the mean value of the noise term to the mean of the raw signal. The behavior of the simulator was analyzed for every one of the resulting 120 scenarios and compared to the original performance.

### 5.2 Design

Four reactive intentions were designed that encapsulate the possible interaction actions that the robot can generate, namely, looking around , following the human face , following the salient object in the environment, and looking at the same place the human is looking at. Each one of those intention is implemented as a simple state machine (Mohammad et al., 2007). The sufficiency of those intentions was based on the fact that in the current scenario the robot simply has no other place to look, and the necessity was confirmed empirically by the fact that the three behavioral processes needed to adjust the intentionality of all of these intentions.

The analysis of natural listening requirements showed the need of three behavioral processes. Two processes to generate an approach-escape mechanism controlling looking toward the human operator which is inspired by the *Approach-Avoidance* mechanism suggested in (Argyle, 2001) in managing spatial distance in natural human-human situations. These processes were named Look-At-Human, and Be-Polite. A third process was needed to control the realization of the mutual attention behavior. This process was called Mutual-Intention. The details refer to (Mohammad et al., 2007). A brief description of them is given here:

1. *Look-At-Human*: This process is responsible of generating an attractive virtual force that pulls the robot's head to the location of the human face.

2. *Be-Polite*: This process works against the *Look-At-Human* process generating a repulsive virtual force that pulls the robot's head away from the location of the human face depending on the period of attending to the human.

3. *Mutual-Attention*: This process tries to pull the robot's head toward direction to which the human is looking.

Table 2: Comparison between the Simulated and Natural Behavior.

| Item | Statistic | Simulation | H-H value |
|---|---|---|---|
| Mutual Gaze | Mean<br>Std.Dev. | 31.5%<br>1.94% | 30%<br>– |
| Gaze Toward Instructor | Mean<br>Std.Dev. | 77.87%<br>3.04% | 75%<br>– |
| Mutual Attention | Mean<br>Std.Dev. | 53.12%<br>4.66% | unknown<br>– |

Five perception processes were needed to implement the aforementioned behavioral processes and intentions. The details can be found in (Mohammad et al., 2007).

### 5.3 Results

Some of the results of numerical simulations of the listening behavior of the robot are given in Table 2. The table shows the average time of performing two basic interactive behaviors obtained from the simulated robot in comparison to the known average values measured in human-human interaction situations. The average times in the human-human case are reported from (Argyle, 2001). As the table shows the behavior of the robot is similar to the known average behavior in the human-human case for both mutual gaze and gaze toward instructor behaviors and the standard deviation in both cases is less than 7% of the mean value which predicts robust operation in real world situations.

Although the average time of showing the three evaluation behaviors are similar to the human-human values as shown in Table 2, this similarity is not enough for completely judging human like natural behavior and the dynamical aspects of the interaction must be taken into account before drawing final conclusions. This method of analysis, although limited, was selected because of two reasons. First the behavior of the speaker in this experiment is not totally natural because the robot did not respond at real time, which affects the dynamics of the interaction but it was hypothesized that the effect on the averages used for evaluation is much less. Second there is no available data about the detailed dynamics of the behavior of the listener and the speaker in the human-human case during explanation scenarios. In near future a wide scale human-human experiment will be conducted by the authors to collect such data for more accurate evaluation.

Fig. 2 shows the effect of increasing the error level on the percentage of time mutual gaze and gaze to-

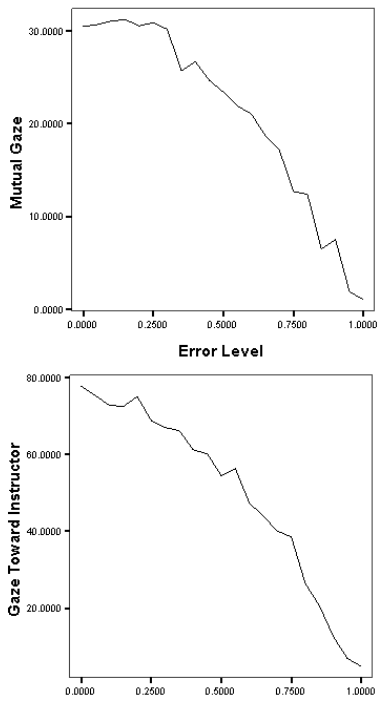


Figure 2: Effect on the error level on the behavior of the robot.

ward instructor behaviors were recognized in the simulation. As expected the amount of time spent on these interactive behaviors decreases with increased error level although this decrease is not linear but can be well approximated with a quadratic function as regression analysis revealed. This means that the performance degrades *gracefully* with the increased noise level even for signal correlated noise.

## 5.4 Discussion

One of the main purposes of having robotic architectures is to make it easier for the programmer to divide the required task into smaller computational components that can be implemented directly. The proposed action integration mechanism helps in achieving a *natural* division of the problem by the following simple procedure. First the task is analyzed to find the basic competencies that the robot must possess in order to achieve this task. Those competencies are not complex behaviors like *attend-to-human* but finer behaviors like *look-right*, *follow-face*, etc. Those competencies are then mapped to the *intentions* of the system. Each one of these intentions should be carefully engineered and tested before adding any more components to the system

The next step in the design process is to design the behavior level integration part of the action integration. To do that, the task is analyzed to find the underlying processes that control the required behavior. Those processes are then implemented. The most difficult part of the whole process is to find the correct parameters of those processes to achieve the required external behavior. Currently this parameter choice is done using trial-and-error but it will be more effective to use machine learning techniques to learn those parameters from the interactions either offline or online. The current architecture supports run-time adaptation of those parameters, and this feature will be exploited in the future to implement learning of the behavioral integration layer. Those behavioral steps are added incrementally and the relative timing between them is adjusted according to the required behavior.

This simple design procedure is made possible because of the separation between the basic behavioral components (intentions) and the behavior level integration layer (processes).

It is informative to compare this procedure with the procedure suggested in (Ishiguro et al., 1999) for the situated modules architecture. Every situated module should have a list of *preconditions* that is always checked by the module controller which chooses that module that is most suitable to the situation. The problem with this arrangement for HRI applications is that the evaluation of the preconditions can be very time consuming or even very difficult to decide in the first place. Let's consider the situated module *look-at-human* that should be activated during the interaction enough to make the speaker feel comfortable but not too much. How can the designer find all the rule to select all the occasions in which the behavior is to be invoked? and how can this list be updated for reuse in other applications? The main problem is that it is too difficult to find the preconditions for a behavior as simple as *look-at-human* and the only solutions are either to complicate the behaviors used (increase the granularity) or to complicate the module controller (may be by using deliberation). In the proposed system this problem does not exist because the behavior itself (the intention) is coded without any need to think about its preconditions. The behavior level integration processes are then designed based on the global view of the task and not the requirements of each intention which means that this set of preconditions need not be built at any point in the design process. This allows intentions to be thinner than the situated modules without the need of a higher deliberative layer.

A widely accepted definition of intention is: "*a choice with commitment*" (Cohen and Levesque,

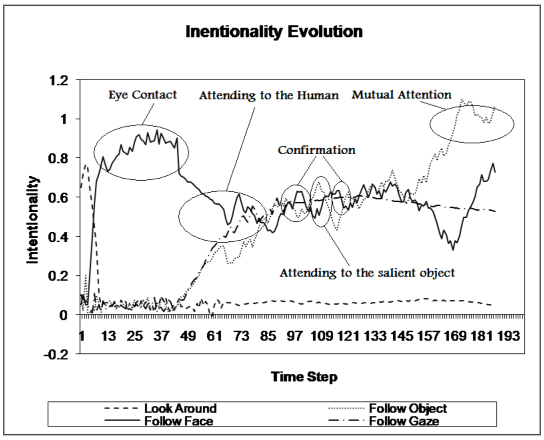


Figure 3: The evolution of intentionality of the four basic intention used in the listener robot.

1990). This definition emphasizes that for a behavior to be perceived as intentional it must possess a form of behavioral *inertia* that makes the robot stick with its behavior until its goal is achieved or the situational changes result on the adoption of a new goal. Careful design of the behavior level integration processes can achieve this goal in the proposed architecture. For example Fig. 3 shows the evolution of intentionality of the aforementioned four basic reactive intentions in one case annotated with the external behavior as perceived by the interacting human. As shown in the figure the effects of the fluctuations of the input signals ,although mapped to the intentionality of various intentions, do not affect the final behavior directly because of the existence of the central action level integrator that averages those fluctuations and provides this needed *behavioral inertia* to make the final behavior look intentional for the human.

# 6 CONCLUSIONS

This paper analyzed the required properties for action integration mechanisms suitable for interactive robots, and presented the design of a new two layers hybrid action integration mechanism that utilizes both distributive integration in the behavior integration layer and a central action level integrator that can achieve the aforementioned requirement. The paper also presented a simple example of a listener robot implemented based on the proposed mechanism.

# REFERENCES


Argyle, M. (2001). *Bodily Communication*. Routledge; New Ed edition.

Arkin, R. C. (1993). *Biological neural networks in invertebrate neuroethology and robotics*, chapter Modeling neural function at the schema level: Implications and results for robotic control, page 17. Academic Press.

Breazeal, C. Kidd, C. T. A. H. G. B. M. (2005). Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005)*, pages 708–713.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23.

Cohen, P. and Levesque, H. (1990). Intention is choice with commitment. *Artificial Intelligence*, 42:213–261.

Ishiguro, H., Kanda, T., Kimoto, K., and Ishida, T. (1999). A robot architecture based on situated modules. In *IEEE/RSJ Conference on Intelligent Robots and Systems 1999 (IROS 1999)*, volume 3, pages 1617–1624. IEEE.

Kanda, T., Kamasima, M., Imai, M., Ono, T., Sakamoto, D., Ishiguro, H., and Anzai, Y. (2007). A humanoid robot that pretends to listen to route guidance from a human. *Autonomous Robots*, 22(1):87–100.

Maes, P. (1989). How to do the right thing. *Connection Science*, 1(3):291–323.

Mohammad, Y. F. O. and Nishida, T. (2007). A new, hri inspired, view of intention. In *AAAI-07 Workshop on Human Implications of Human-Robot Interactions*, pages 21–27.

Mohammad, Y. F. O., Ohya, T., Hiramatsu, T., Sumi, Y., and Nishida, T. (2007). Embodiment of knowledge into the interaction and physical domains using robots. In *International Conference on Control, Automation and Systems*, pages 737–744.

Nicolescu, M. N. and Matarić, M. J. (2002). A hierarchical architecture for behavior-based robots. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 227–233, New York, NY, USA. ACM.

Perez, M. C. (2003). *A Proposal of a Behavior-based Control Architecture with Reinforcement Learning for an Autonomous Underwater Robot*. PhD thesis, University of Girona.

PhaseSpace (2007). http://www.phasespace.com/.

S Karim, L Sonenberg, A. T. (2006). A hybrid architecture combining reactive plan execution and reactive learning. In *9th Biennial Pacific Rim International Conference on Artificial Intelligence (PRICAI)*.

Steels, L. (1993). *The artfcial route to artfcial intelligence. Building situated embodied agents*, chapter Building agents with autonomous behaviour systems. Lawrence Erlbaum Associates, New Haven.