Tradeoffs in Making the Behavior-Based Robotic Systems Goal-Directed

Amol Dattatraya Mali
Dept. of computer science and engineering
Arizona state university, Tempe, AZ 85287-5406, USA
amol.mali@asu.edu

Abstract

[1] A number of behavior-based autonomous robots with varying degree of reactive functionality have been built. The principle of avoiding an explicit representation of goals inside these robots has limited their achievements to navigation. As a result reactive functionality and goal fulfilling capability have remained orthogonal. The current autonomous robot architectures focus extensively on modifying the internal computational mechanisms of robots, ignoring the important issue of the tradeoffs in robot-environment-goal fit. We provide results on the latter issue. We develop a model of behaviors, environmental modification and goals. Defining the notion of coupling that captures dependency within the internal structure of a behavior space, we show that more complex goals demand higher coupling or more behaviors or a modification to the environment. We show that making behavior-based robots goal-directed without introducing a planner and explicitly represented goals can also mean making the environment goal-directed or decreasing the modularity of the behavior space by increasing the coupling. We argue that these tradeoffs show a spectrum of architectures for integrating goals and behavior-based functionality.

1 Introduction

Artificial Intelligence paradigms today are moving towards a more distributed agent-based architecture. It is argued that when intelligence is approached in such an incremental manner, reliance on representation disappears [3][4]. Early research on robots assumed environment to be completely static, predictable and avoided sensory processes, e.g. in the first robot project [13], Shakey executed simplified plans and manipulated blocks, with full a priori knowledge of the types of objects etc. Internal representations were considered to be responsible for the failure of this approach in more complex environments. Ideas of Brooks [3] lead researchers to build robots without global representations and hierarchy. Robots in such systems (also called *situated robots*) are largely reactive and communicate though the world by making changes to it that other robots can perceive. It is argued that intelligence emerges from the interaction of the robots with the world since such an evidence has been found in social insects like ants, wasps and bees. Hence traditional centralized planning is replaced by robot-environment dynamics.

Since some such reactive robots exhibited problems like deadlocks and myopic functionality, hybrid architectures with a deliberative component to fix these problems as in [2] began to be explored. The previous research on autonomous robots has concentrated on adding separate modules to make behavior-based robots goal-directed. The conventional wisdom is - autonomous robot architectures should be tiered and the lack of tiering places a serious limitation. We review this argument in our research here.

We explore the option of modifying the behavior structure and environment for fulfilling goals, rather than introducing additional modules like a planner and/or a sequencer. We show that an environment has an important role to play in increasing the goal-directedness of the situated robots. The notation developed here and the analysis provided are intended to reveal the tradeoffs in making the behavior-based robots goal directed. We examine the relation between the dependency within a behavior space (defined in terms of *coupling* in section 2), goals that can be fulfilled by it and its environment. We discuss the mechanisms of externalizing internal states that enhance reactivity. We show how an increase in the complexity of goals affects a behavior space and its environment. These tradeoffs indicate a new spectrum of architectures for integrating goals and reactive functionality.

2 A Model of Behavior-based Interaction

We develop a model of behaviors, goals and environmental modification and use it in section 3 to prove

[1]Appears in the proceedings of ICRA-1998 (IEEE International Conference on Robotics & Automation, Belgium, pp. 1128-1133).

our results.

• **Behavior** - A behavior β_i is ith behavior of a robot, where $1 \leq i \leq \mid B \mid$, modeled as a 2 tuple $< s_i, c_i >$ and defined as a mapping from stimulus s_i to consequence c_i, both expressed in CNF (conjunctive normal form). B denotes the set of all behaviors of a robot, also called as behavior space. Note that though this representation of behaviors is same as the $< pre-condition, post-condition >$ representation of state transforming actions or operators in the planning literature, the execution criteria for behaviors and operators are different. The behavior of an autonomous robot is generally executed once the stimulus is true, while an operator is executed only if its pre-conditions are true and the operator is relevant to the goal that the planner is trying to achieve. Purely behavior-based autonomous robots do not have an explicit representation of goals.

• **Stimulus** - It is assumed that each literal in a stimulus corresponds to a number of sensor readings, i.e. sensor readings are processed to extract meaning out of them and there may be a literal to which this meaning is mapped. For example, if readings of 10 sonars are all less than a certain limit, a wall or a big obstacle may be located nearby. Then some literal in stimulus for the behavior *avoid_obstacle* will become true. If there are p sensors, each of which can have m distinct readings, we do not consider them to be m^p distinct stimuli, since all these readings can be mapped to fewer literals. s_i may contain implications of the form

$$(\bigwedge_{m=1}^{k} A_m \Rightarrow \bigwedge_{s=1}^{w} P_s)$$

(discussed in the definition of coupling in this section), to model parameter passing among behaviors that is necessary to make the construction of certain behavior chains (defined in this section) and hence fulfillment of certain goals possible (A_m, P_s are predicate symbols). Such an implication can be converted into clauses, thus maintaining s_i in CNF. The stimulus s_i is defined to be at least as *strong* as stimulus s_j if $(s_i \Rightarrow s_j)$. Stimuli for default behaviors like random wandering are assumed to be of the form $\neg(a_1 \wedge a_2 \wedge a_3 .... \wedge a_h)$. This means that the robot wanders as long as a certain condition is false.

• **Behavior chain** - Complex behavior occurs because a number of primitive behaviors (e.g. β_i) operate sequentially and/or concurrently. Here we focus on the temporal sequencing mechanism that gives rise to a complex behavior. A behavior chain C is a temporal sequence of behaviors $\{\beta_{i_1} : \beta_{i_2} : \beta_{i_3} : ... : \beta_{i_k}\}$, where $1 \leq i_m \leq \mid B \mid$, $1 \leq m \leq k$. $\beta_{i_m} : \beta_{i_{m+1}}$ is used to denote that these two behaviors occur immediately next to each other in time, with the former behavior preceding the latter. Such a chain is said to be composable from B (denoted by $C \lhd B$) if behaviors in the chain are elements of B. All possible temporal chains of behaviors are not programmed a priori, they get composed in real time in a situation driven manner. The action of an earlier behavior in the chain changes the situation in such a way that the newly changed part of the situation in conjunction with the universe U (unaffected part of the situation) implies stimulus for the next behavior. Consider the chain $\{\beta_1 : \beta_3 : \beta_7\}$. Here we may have $c_1 = (a \bigvee b) \bigwedge (c \bigvee d)$ and $s_3 = (a \bigvee b \bigvee e) \bigwedge (c \bigvee d \bigvee f) \bigwedge z$. Here $(c_1 \Rightarrow s_3)$ if $U \Rightarrow z$. z is not explicitly listed in c_1.

• **Goal Complexity** - A primitive goal g_i is considered to be specified as a 2-tuple $< I_i, F_i >$, where I_i and F_i denote the initial and final states of object i respectively. These are assumed to be expressed in purely conjunctive form. A primitive goal is considered to be more complex if it is to be achieved from a *weaker* initial state and should lead to a *stronger* final state. A goal g_i is said to be at least as complex as goal g_j if $(I_j \Rightarrow I_i) \bigwedge (F_i \Rightarrow F_j)$. A non-primitive goal G consists of a set of such primitive goals. A set of goals G' is defined to be more complex than a set G (denoted by $G' >_c G$) if G' is obtained from G by replacing one or more primitive goals $g_i \in G$ by more complex goals and/or adding more goals to G.

• **Marker** - We use the term marker to refer to ((a) new objects introduced in an environment or (b) new features added to current objects in an environment or (c) those features of current objects that were not used before but used later), with the intention of externalizing internal state of a behavior e.g. if a robot is supposed to collect all tennis balls except those near a cupboard, one way to design the stimulus is to store the absolute location of the cupboard in the form of internal state and design *pickup* behavior of the robot not to pick up balls within some radius around that location. However one can install a red pole near the cupboard and replace the absolute location of the cupboard in the stimulus of the behavior by presence of red pole that can be sensed by vision. The red pole is a marker. Markers can serve other important purpose besides externalizing internal state, e.g. as a tool for dealing with noisy sensors and reducing perceptual computations. But these uses of markers are not considered here.

A marker is denoted by M_i and is described by a conjunction of its features. For example, a colored cube kept on a flat surface can serve as a marker and be described as

$\exists x \exists y_1, y_2 (cube(x) \bigwedge is\_face\_of(x, y_1) \bigwedge color(y_1, red)$

$\bigwedge is\_face\_of(x, y_2) \bigwedge color(y_2, green))$. A marker M_i is at least as *strong* as a marker M_j if $(M_i \Rightarrow M_j)$. It is *stronger* if $(M_i \Rightarrow M_j)$ is a tautology and the set of predicates in M_j is a proper subset of the predicates in M_i. A set of markers becomes *stronger* as more markers are added to it and/or existing markers from that set are replaced by *stronger* ones. The relation *stronger* is intended to indicate that the stronger marker set can be used in at least as many ways as the weaker marker set, for the purpose of externalizing internal states. We assume that an addition of markers to an environment does not destroy or hide the stimuli.

• **Degree of Reactivity** - We are interested in the reactivity along spatial dimension - the amount of internal state. Some internal states have to be updated whenever external world changes. Externalizing states eliminates such updates since the most recent information is available in the world itself. Hence there are reasons for a robot to be more reactive. Let the set of predicates occurring in stimulus s_i of a behavior β_i be S_i. Let the set of those predicates from S_i whose truth value is decided by the robot based on external information gathered by sensors be S_{ir}. The degree of reactivity of the behavior is said to increase if S_{ir} is modified to S'_{ir} such that $S_{ir} \subset S'_{ir}, S'_{ir} \subseteq S_i$. The definition of degree of reactivity is independent of the time required to sense or deliberate to determine the truth of predicates and is concerned only with the spatial dimension corresponding to internal storage.

• **Environment -** $E \propto E'$ denotes that the environment E' is obtained from the environment E by adding zero or more markers to E and/or replacing the existing markers by *stronger* ones.

• **Coupling** - Coupling c_{im} is said to exist between two behaviors β_i and β_m if values of some variables in some literals in s_m are set by conditions of the form $(\bigwedge_{t=1}^{k} A_t \Rightarrow \bigwedge_{s=1}^{w} P_s)$ (explained later here) in s_i and is defined as the function $f(k, u)$ (f is such that $f(k, u) \geq f(k', u')$ if $k \geq k', u \geq u'$ and $f(k, u) > f(k', u')$ if $(k > k', u \geq u')$ or $(k \geq k', u > u')$, $f(0, 0) = 0$) where k is the number of literals in s_m, one or more variables of which are assigned values by conditions in s_i and u is the number of literals in the corresponding conditions in s_i, e.g. if a robot is to place painting brushes near windows, then there may not be any coupling between the behavior that picks up a painting brush (say β_1) and the behavior that puts it near a window (say β_2). In that case $s_1 = \exists x(graspable(x) \bigwedge brush(x))$ and $s_2 = \exists x, y(in\_hand(x) \bigwedge brush(x) \bigwedge robot\_at(y) \bigwedge window(y))$. If it is desired that big brushes should be kept near big windows and small brushes near small windows, s_1 can be modified to be $s_1 = \exists x(graspable(x) \bigwedge brush(x) \bigwedge (small(x) \Rightarrow assign(Y, sm)) \bigwedge (big(x) \Rightarrow assign(Y, bg)))$ where it is assumed that in evaluating the truth of the condition $(A_m \Rightarrow P_s)$ (where A_m is expressed in conjunctive form and P_s is an atomic formula or a conjunction of atomic formulas of arity 2, of type $assign(x, a)$, introduced for the purpose of making assignments), P_s is evaluated to be true only if A_m is true and $assign(x, a)$ sets value of the variable x to a. s_2 can be modified to be $\exists x, z(in\_hand(x) \bigwedge brush(x) \bigwedge robot\_at(z) \bigwedge window(z) \bigwedge size(z, Y))$. Here β_1 and β_2 are behaviors that are coupled. The clauses involved in coupling are $(small(x) \Rightarrow assign(Y, sm)), (big(x) \Rightarrow assign(Y, bg))$ and $size(z, Y)$. Hence the actual coupling c_{12} is $f(1, 4)$. The coupling C of a behavior space increases when coupling c_{ij} between some behavior pair β_i, β_j increases, couplings between other pairs staying constant. The coupling captures dependency among stimuli of behaviors and the modifications made to their structure to fulfill a goal. Coupling, marker strength and goal complexity are relative measures.

3 Effects of Increasing Goal-Directedness

In this section, we discuss our results on relations between environment, complexity of goals and coupling.

Theorem 1. As the complexity of a goal increases, the coupling of the behavior space changes from C to C' where $C' \geq C$, when $\mid B \mid, E$ are left unchanged.

Proof - Let the goal G' be derived from the goal G by adding tuples of the form $< I_i, F_i >$ to G, and/or replacing existing goals by more complex goals, so that $G' >_c G$.

When $\mid B \mid$ is kept constant and it is desired that goals not currently fulfilled by the system should be fulfilled, the only option is to modify the stimuli of existing behaviors so that they are chained in a certain way to fulfill the more complex goal. We consider how the stimuli of behaviors in the current system can be modified to fulfill the goals $g \in (G' - G)$, without dropping any literal l from existing stimuli (since dropping literals from existing stimuli to fulfill $(G' - G)$ may result in some $g_i \in G$ not being fulfilled).

Case 1. Consider behaviors β_i and β_m where s_i contains conditions that assign values to variables in literals of s_m. One can modify the values that are being assigned in $assign(x, a)$ to fulfill elements of $(G' - G)$ or rearrange existing literals, e.g. if instead of dropping big brushes at big windows and small brushes at small windows, it is desired that big brushes should be dropped at small windows and small brushes should be dropped at big windows, the assignment conditions $(small(x) \Rightarrow assign(Y, sm))$, $(big(x) \Rightarrow assign(Y, bg))$ in s_i can be changed to $(small(x) \Rightarrow assign(Y, bg)), (big(x) \Rightarrow$

$assign(Y, sm)$). If it is desired that small brushes should not be moved, s_i can be changed to include $(small(x) \Rightarrow assign(Y, sm)) \bigwedge \neg small(x)$. However these changes leave k, u (in the definition of coupling) unchanged. This argument can be repeated for other pairs of stimuli. Hence $C' = C$.

Case 2. Stimuli are changed by adding new conditions to s_i and/or increasing the number of literals in s_m, variables of which are assigned values. This leads to an increase in k and/or u. Let k, u be increased by k' and u' respectively. In that case, the new coupling between the behaviors is $f(k+k', u+u') > f(k, u)$. For example, if in the goal of moving brushes, there is a third category of brushes and windows, say *medium*, then an additional condition $(medium(x) \Rightarrow assign(Y, md))$ will be needed. Here u is increased. If it is required that brushes of a particular size should be dropped at only those windows which also have a cupboard of corresponding size near them, stimuli will have to be modified, increasing both k and u. This argument can be extended to multiple pairs of behaviors. Since the coupling of at least one behavior pair increases, $C' > C$. Hence the proof. □

Only one robot is responsible for moving and dropping the brush at desired location, once it picks up a brush in the example of moving brushes discussed above. To eliminate coupling, one will have to design multiple behaviors for picking and dropping, e.g. separate behaviors for picking brushes of different sizes with stimuli $\exists x(small(x) \bigwedge brush(x))$, $\exists x(big(x) \bigwedge brush(x))$ and $\exists x(medium(x) \bigwedge brush(x))$ and corresponding behaviors for dropping the brushes. Then even if a robot picks up a brush and drops it at an incorrect location, other robot can pick it up and drop at the correct location independently. This suggests another dimension of analysis.

Theorem 2. If the coupling of a behavior space is changed from C to C', $C' < C$, keeping $\mid B \mid$ fixed, fulfilling original goals requires introduction of m markers, $m \geq 1$.

Proof - Since coupling of the space is reduced, there exists a pair of behaviors, β_i, β_m, such that the coupling c_{im} between them is reduced. This means that either the number of literals in s_m, variables of which were assigned values by conditions in s_i were reduced and/or the number of literals in conditions in s_i making truth assignments were reduced (the coupling c_{im} reduces in these cases). These changes will either result in variables in literals in s_m not having any values assigned to them or variables that have some value assigned arbitrarily(these values are not set by conditions in s_i, e.g. instead of letting a condition in s_i set value of Y in $at(Y)$ in s_m, one can force it to some arbitrary value and reduce the coupling) or variables that have incorrect values assigned, resulting in fewer goals fulfilled or an undesired behavior. One has to then create markers that act as substitutes for the values to be assigned, (e.g. one can write the size of window near which a brush is to be dropped on the brush itself and then modify the previously discussed stimuli $s_1 = \exists x(graspable(x) \bigwedge brush(x) \bigwedge (small(x) \Rightarrow assign(Y, sm)) \bigwedge (big(x) \Rightarrow assign(Y, bg)))$, $s_2 = \exists x, y(in\_hand(x) \bigwedge brush(x) \bigwedge robot\_at(y) \bigwedge window(y) \bigwedge size(y, Y))$
to $s_1 = \exists x(graspable(x) \bigwedge brush(x) \bigwedge has\_word(x))$ ($has\_word(x)$ means that x has a word written on it) and $s_2 = \exists x, y, z(in\_hand(x) \bigwedge brush(x) \bigwedge robot\_at(y) \bigwedge window(y) \bigwedge on(x, z) \bigwedge word(z) \bigwedge size(y, z))$. This introduces $m > 0$ markers, hence the proof. □

From Theorem 1, Theorem 2 and the definition of environmental transformation, we have

Theorem 3. When a goal G to be fulfilled by a behavior space in environment E is modified to more complex G' ($G' >_c G$) and the coupling C and $\mid B \mid$ are kept constant, fulfilling G' requires modification of E to E', such that $E \propto E'$.

This result also shows that making a behavior-based robotic system goal-directed may also mean making its environment goal directed, since markers are seeded in the environment to bias the behaviors towards fulfilling goals.

4 Behavior-based House Cleaning

We discuss here how the ideas presented before can be used. Consider a robot with following behaviors for house cleaning - $pick\_air\_freshener(\beta_1), drop\_air\_freshener(\beta_2)$, $spray\_air\_freshener(\beta_3), wander(\beta_4)$, $pick\_floor\_mop(\beta_5), sweep\_floor(\beta_6)$, $drop\_floor\_mop(\beta_7), pick\_bowl\_scrub\_brush(\beta_8)$, $drop\_bowl\_scrub\_brush(\beta_9), scrub\_bowl(\beta_{10})$, $pick\_tub\_scrub\_brush(\beta_{11})$, $drop\_tub\_scrub\_brush(\beta_{12})$, $scrub\_tub(\beta_{13})$. Consider the goal of sweeping the floor. The chain $\{\beta_5 : \beta_6 : \beta_7\}$ fulfills this goal. Once the robot picks up the mop, it should sweep the floor. However it is possible that it will drop the mop immediately, since having the mop in hand provides the stimulus for dropping it. To increase the probability of execution of the chain $\{\beta_5 : \beta_6\}$, we can couple the behaviors β_5 and β_4, so that β_5 sets values of variables in literals in the stimulus of *wander* to wander till dirty floor is found. Once a dirty floor is found, the stimulus of β_6 will be triggered and the floor will be swept.

If one decides not to introduce the coupling, one can introduce a marker in the environment and design the stimulus of the behavior *wander* to move the robot till the marker is found. It is possible that the mop will be dropped at an undesired place after the floor is swept. To prevent that, one can introduce a different wander behavior to ensure that the robot moves to the correct place for dropping the mop. One can avoid the introduction of the new *wander* behavior by introducing coupling or adding more markers. The chain $\{\beta_1 : \beta_3 : \beta_2\}$ freshens air. The chain $\{\beta_8 : \beta_{10} : \beta_9\}$ cleans toilet bowl. The chain $\{\beta_{11} : \beta_{13} : \beta_{12}\}$ fulfills the goal of cleaning the bath tub. To reliably compose these chains, one can introduce coupling and/or markers, as discussed in the case of sweeping the floor. One may be required to prioritize the behavior modules to resolve conflicts.

5 Discussion

We review some architectures of autonomous robots in this section and show how our approach differs from them.

Basic behaviors for navigation such as "avoiding obstacles", "moving towards free space", "following targets", "follow wall", "explore open space", "track target" and "search target" are implemented on robots [10]. There is no integration of more complex goals in the architecture. Gat & Dorais [9] propose conditional sequencing as a mechanism for navigation. A conditional sequence is similar to a script. They have implemented navigation behaviors like "dead reckoning", "avoiding obstacles", "following walls". Saffiotti et al [14] argue that engaging in more purposeful activities than wandering requires more than pure reactivity and that an explicit reasoning to fulfill goals is needed. Their work fits in the tradition of the "two level" approaches to robot control in which a strategic planner is used to guide a reactor. Simmons' methodology in the task control architecture [16] is to first develop systems having sequential sense-plan-act cycles and then use additional facilities to add concurrency. He says that the decision making for autonomous robots should be a combination of both reactivity and planning. Connell [5] proposes a three layer architecture with symbolic, subsumption and servo layer (which directly interacts with the world). The symbolic layer maintains a world model and takes strategic decisions. Lyons & Hendriks [11] advocate a combination of the ability to plan and react. Their planner maintains a world model and tunes the behavior of reactor to achieve goals. Fujimura [7] presents a distributed approach for multi-robot navigation. Each robot plans reactively and the plans may be aborted, revised or completed, depending upon the situation. This is similar to a planner-reactor style hybrid architecture.

Behavior hierarchies are proposed by Seeliger & Hendler [15] to avoid explicit planning. However these hierarchies provide a detailed guidance for achieving the goals. Such hierarchies are provided by a domain expert and may not always be available. The most abstract modules lie at the root of behavior hierarchy and they are triggered by an explicitly stated goal rather than environmental stimuli. Maes [12] proposes an algorithm for autonomous action selection that combines the characteristics of both traditional planners and reactive systems. This algorithm makes it possible for one to trade off goal-orientedness for situation-orientedness, bias towards ongoing plans (inertia) for adaptivity, thoughtfulness for speed and adjust sensitivity to goal conflicts. However the activation from goals and the current situation is computed and spread in an ad-hoc manner, making the outcome unpredictable.

Behaviors like "avoid obstacle", "decelerate at curve", "decelerate at narrow road", "wander","follow left edge", "follow right edge", "follow road" are implemented using potential fields in [17]. Huber & Grupen [18] propose a hybrid discrete event dynamic systems approach to robot control in which reactive module and a symbolic reasoner to generate control strategy are used. Ferguson [19] proposes a hybrid architecture combining deliberative and reactive control.

It is true that autonomy does not necessarily entail an extreme level of intelligence. For example, an ant is autonomous, though it does not exhibit higher level of intelligence. Though an automated chess player exhibits higher level of intelligence, it is not autonomous. As a result, it is the criterion of autonomy rather than intelligence, that was used to evaluate the success of the early implementations of behavior-based agency. Explicit goals and reasoners were added on the top of behavior spaces, since goal fulfilling functionality did not always emerge from the robot-environment interaction.

Though only a few hybrid architectures have been quoted here, the actual list is very long. Gat [8] argues that it is easy to combine different computational mechanisms, and hence an architecture should be heterogeneous. He describes a hybrid architecture that integrates classical world models with reactive systems. Despite such claims, the division of functionality between reactor and planner remains a matter of considerable debate. Does learning overcome these limitations ? A closer look at the learning in situated robots shows reinforcement learning to be the most widely used form of learning. Dorigo & Schnepf [6] report a simulated robot that learns to follow light and avoid hot dangerous objects. This however turns out to be an even more challenging task, since both the behaviors as well as the co-ordination among them has to be learnt (whereas reactors have a ready to use set of behaviors and planners have pre-canned co-ordination strategies). Though learning is more suitable for domains where such pre-canned sets are not available, the difficulty in specifying the right reward and punishment functions has placed a limitation on the success of these architectures. We do not claim that our approach of introducing coupling and/or markers is an alternative to learning. However our approach does address the problem of orthogonality of reactive and goal fulfilling functionality, without introducing heterogeniety, an issue that the architectures of learning for situated robots are struggling with.

Clearly, higher coupling between behaviors makes a behavior space less modular. Instead of using a planner and/or sequencer to compose goal fulfilling behavior chains, we modify the $< s_i, c_i >$ type behavior structure itself and/or the environment. Such a structural modification increases the probability of goal fulfillment and makes the behavior-based robots more goal directed. We do not argue that such modifications can always replace an explicit planner. However they challenge the belief that reactive behaviors need explicit planners for tasks more complex than simple navigation. Planners in the current hybrid architectures fulfill goals at the cost of reactive functionality. Since the introduction of coupling and markers does not explicitly select and order behaviors to achieve explicit goals, our approach is different from the previous architectures.

6 Conclusion

Our results apply to other kinds of autonomous agents as well, e.g. some search algorithms of a softbot trying to retrieve data fulfilling certain characteristics from a database can be viewed as its sensors and markers can be introduced in the database to assist the search algorithms. Balch and Arkin [1] describe reactive robots that use potential fields to compute behavioral response. Even if our first order logic based $< s_i, c_i >$ representation is changed to some other, say potential fields (which considers stimulus to be a summation of potential fields that are attractive or repulsive, and hence stimulus there is a grey level value rather than boolean), our arguments about relations between coupling, goal fulfilling capability and environmental transformation continue to apply.

We showed that more complex goals increase coupling and that trying to eliminate coupling increases the number of required behaviors or markers. Hence it can be concluded that fulfilling more complex goals while curbing the coupling either makes B more complex to debug (due to increase in the size) or makes the environment more goal directed, requiring reasoning about the impact of markers on overall functionality of the robot. The size of a behavior space, environment and the coupling have an impact on the amount of testing required to verify that the robot will exhibit an acceptable level of reactive and goal fulfilling functionality. The environment affects the perceptual computations and the goal complexities are related to a customer's expectations from the robot. The tradeoffs we showed indicate a rich space of behavior-based architectures that no longer require reactivity and goal fulfilling capabilities to be orthogonal and provide flexible options for integrating goals with behaviors, maintaining the homogeniety of the computational structure.

References

[1] **Tucker Balch & Ronald C. Arkin**, Communication in reactive multi-agent robotic systems, Autonomous robots 1, 1994, 27-52.

[2] **R. Peter Bonasso and David Kortenkamp**, Characterizing an architecture for intelligent, reactive agents, Working notes of AAAI spring symposium on lessons learned from implemented software architectures for physical agents, 1995, 29-34.

[3] **Rodney A. Brooks**, Achieving artificial intelligence through building robots, AI memo 899, MIT, May 1986.

[4] **Rodney A. Brooks, Jonathan H. Connell and Peter Ning**, HERBERT: A second generation mobile robot, AI memo 1016, MIT, Jan. 1988.

[5] **Jonathan H. Connell**, SSS: A hybrid architecture applied to robot navigation, IEEE international conference on robotics and automation, 1992, 2719-2724.

[6] **Marco Dorigo and Uwe Schnepf**, Genetics-based machine learning and behavior-based robotics: A new synthesis, IEEE transactions on systems, man and cybernetics, Vol. 23, No. 1, Jan/Feb. 1993, 141-154.

[7] **Kikuo Fujimura**, A model of reactive planning for multiple mobile agents, Procs. of the IEEE international conference on robotics and automation, 1991, 1503-1509.

[8] **Erann Gat**, Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real world mobile robots, Procs. of AAAI-92.

[9] **Erann Gat and Greg Dorais**, Robot navigation by conditional sequencing, Proceedings of the IEEE conference on robotics and automation, 1994.

[10] **I. Kweon, Y. Kuno, M. Watanabe and K. Onoguchi**, Behavior-based mobile robot using active sensor fusion, Proceedings of IEEE international conference on robotics and automation, 1992, 1675-1682.

[11] **D. M. Lyons and A. J. Hendriks**, Planning for reactive robot behavior, Proceedings of the IEEE international conference on robotics and automation, 1992, 2675-2680.

[12] **Pattie Maes**, How to do the right thing, AI memo 1180, MIT, Dec. 1989.

[13] **Nilsson, Nils**, "Shakey the Robot", SRI AI Center Technical Note 323, April 1984.

[14] **Alessandro Saffiotti, Enrique Ruspini and Kurt Konolige**, Blending reactivity and goal-directedness in a fuzzy controller, Procs. of second IEEE conference on fuzzy systems, 1993, 134-139.

[15] **Oliver Seeliger and James Hendler**, Supervenient hierarchies of behaviors: Lessons learned from a vacuuming robot, Working notes of the AAAI Spring symposium on lessons learned from implemented software architectures for physical agents, 1995, 187-195.

[16] **Reid Simmons**, An architecture for co-ordinating planning, sensing and action, Presented at DARPA Planning workshop, Nov. 1990.

[17] **Wang Tianmiao and Zhang Bo**, Time-varying potential field-based 'perception-action' behaviors of a mobile robot, IEEE international conference on robotics and automation, 1992, 2549-2554.

[18] **Manfred Huber and Roderic A. Grupen**, A hybrid discrete event dynamic systems approach to robot control, Technical report 96-43, Dept. of computer science, Univ. of Massachusetts, 1996.

[19] **Innes Ferguson**, Integrating models and behaviors in autonomous agents: Some lessons learned on action control, Working

notes of the AAAI Spring symposium on lessons learned from implemented software architectures for physical agents, 1995, 78-91.

Acknowledgement - I thank Damian Lyons, Keith Decker, Phil Agre, Paul Rosenbloom, Alessandro Saffiotti, Illah Nourbakhsh, Chitta Baral and Nicholas Kushmerick for their useful comments on the previous draft of this paper.