Constraint-based specification of reactive multi-agent systems

Amol Dattatraya Mali
Dept. of computer science and engineering
Arizona state university, Tempe, AZ 85287-5406
e-mail: mali@tahoma.eas.asu.edu

Abstract

There has been an increasing interest in developing computational theories of agents which are typically reactive. However theoretical foundations of such agents are less developed. It is also not clear what combination of bottom up and top down approaches is the best. We develop a formal model of reactive agency and use it to analyze the interactions among complexities of goals, multi-agent system and its environment. Using the notion of coupling that captures dependency within the internal structure of an agent system, we show that more complex goals demand higher coupling or more behaviors or more complex environment. Visiting Herbert Simon's conjecture that behaviors look complex because they occur in a complex environment, we conclude that complexity of an environment is also related to complexity of goals being fulfilled and in fact, an environment becomes more complex as one tries to externalize internal state with markers. Warning that global effects of adding markers need to be analyzed, we show that behaviors with longer stimuli have the potential to make environments more complex. These results enable us to identify agent-environment-goal tradeoffs. We use these constraints to obtain a specification of multi-agent systems. These architecture independent constraints provide a useful tool for automated agent design.

1 Introduction

Artificial Intelligence paradigms today are moving towards a more distributed agent-based architecture. According to a recent report, the charter for AI in the twenty-first century is to build such agents for use in simulation, human assistance, robot interaction etc. [2]. These agents are typically reactive, treating the world as an external memory from which knowledge can be retrieved just by perception. It is argued that when intelligence is approached in such an incremental manner, reliance on representation disappears [1]. Ideas of [1] lead researchers to build multi-agent systems without global representations and hierarchy. Agents in such systems communicate though the world by making changes to it that other agents can perceive. It is also assumed that behavior emerges from interaction of these "situated" agents with the world.

Much of the previous research on multi-agent systems has concentrated more on awareness of actions of other team members, performance of the systems with change in size of the population and synchronizing agents through communication and less on interactions among complexities of agent systems, goals and the environments. Like us, [4] claims that though reactive agent architectures have been proposed as an alternative to traditional AI, their theoretical foundations are less developed. Externalizing internal states will be indispensable if an agent is to respond in a limited time and if such responses can be obtained faster by extracting information through external world rather than internal states. Some internal states have to be updated whenever external world changes and externalizing states eliminates such updates since most recent information is available in the world itself. Hence there are reasons for an agent to be more and more reactive.

In this paper, we formally probe into the tradeoffs among the dependency within a reactive multi-agent system (defined in terms of *coupling* discussed in section 2), goals that can be fulfilled by it and the complexity of its environment. Making a system more reactive essentially means transforming internal state into external state that can be extracted through perception. Humans do this very often, e.g. instead of memorizing schedules, generally we maintain a copy of them that can be looked up to react appropriately. We discuss the mechanisms of externalizing internal states that enhance reactivity. Then we analyze the effects of increasing the degree of reactivity and show how that affects complexity of an environment. We show how increase in the complexity of goals affects a multi-agent system and its environment. These results indicate the tradeoffs involved in the design of reactive multi-agent systems. In *section 2*, we develop a formal model of reactive multi-agent systems. In *Section*

3 we report our results that throw light on the tradeoffs among complexities of goals, multi-agent systems and environments. We propose constraint-based specification of reactive multi-agent systems in *section 4*. *Section 5* presents our conclusions.

2 A Formal Model of Reactive Agency

Here we develop a formal model of multi-agent systems and use it in section 3 to prove our results. First order logic used here is intended to be only an abstract representation that can be used to expose certain tradeoffs. We do not assume that all real world systems are built with first order logic as their representation. But differences in representations do not affect the applicability of our conclusions to real world systems. It is assumed that a multi-agent system contains n agents and the number of behaviors of agent i is denoted by b_i.

- **Behavior** - A behavior β_{ij} is jth behavior of ith agent, where $1 \leq j \leq b_i$, modeled as a 2 tuple $< s_{ij}, c_{ij} >$ and defined as a mapping from stimulus s_{ij} to consequence c_{ij}.
- **Behavior space** - It is the set of all behaviors of all agents. It is denoted by $B = \{\beta_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq b_i\}$ and $\mid B \mid$ is

$$\sum_{i=1}^{n} b_i$$

If there are four agents in a system with two, three, two and four behaviors respectively, $B = \{\beta_{11}, \beta_{12}, \beta_{21}, \beta_{22}, \beta_{23}, \beta_{31}, \beta_{32}, \beta_{41}, \beta_{42}, \beta_{43}, \beta_{44}\}$.

- **Stimulus length** - It is assumed that stimuli of all behaviors of all agents are expressed in conjunctive normal form (CNF). 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, it may indicate presence of a wall or a big obstacle. The literal in stimulus for *avoid_obstacle* behavior will then correspond to presence of an obstacle, it will not always have sensor readings as arguments. Not every distinct set of sensor readings corresponds to different stimulus. 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. Only those predicates that are required for execution of a behavior are listed in stimulus. Universal truths which are also required for execution of the behavior are not listed since such list can be arbitrarily long. The universe is a conjunction of predicates denoted by U. Hence many times when we say that a stimulus is s_{ij}, we mean that it is $(s_{ij} \bigwedge X)$, where $(U \Rightarrow X)$, X being a part of the universe, e.g. to pick up a can, it is necessary for a robot to have a gripper in a good condition, but this is not listed in stimulus of the behavior *pick_up_can*.

The total number of literals occurring in a stimulus (obtained by counting repeated literals only once and neglecting negated literals if their unnegated versions are present) is defined as stimulus length, denoted by $\mid s_{ij} \mid$, where s_{ij} is stimulus of jth behavior of ith agent. The stimulus s_{ij} is defined to be at least as strong as stimulus s_{xy} if $(s_{ij} \Rightarrow s_{xy})$ and longer if $(\mid s_{ij} \mid > \mid s_{xy} \mid)$. For example, $s_{ij} = a \bigwedge (b \bigvee c) \bigwedge d$ is stronger than $s_{xy} = (b \bigvee c)$, it is also longer since $\mid s_{ij} \mid = 4$ and $\mid s_{xy} \mid = 2$. Stimuli for default behaviors like "wander" that are executed when no other stimulus that can trigger a more important or useful behavior is available are an empty formula ϕ. Such stimuli are considered to be the weakest and the shortest. If $s_{ij} = \phi$, then $\mid s_{ij} \mid = 0$.

- **Consequence length** - It is assumed that consequences of all behaviors of all agents are expressed in conjunctive normal form (CNF). The total number of literals occurring in a consequence is defined as consequence length, denoted by $\mid c_{ij} \mid$ where c_{ij} is consequence of jth behavior of ith agent (obtained by counting repeated literals only once and neglecting negated literals if their unnegated versions are present). Only those predicates whose truth changes as a result of execution of a behavior are listed in its consequence. Hence when we say that consequence of a behavior β_{ij} is c_{ij}, we mean that it is actually $(c_{ij} \bigwedge Y)$, $(U \Rightarrow Y)$. Y can be arbitrarily long and hence is not listed. The consequence c_{ij} is defined to be at least as strong as consequence c_{xy} if $(c_{ij} \Rightarrow c_{xy})$ and longer if $(\mid c_{ij} \mid > \mid c_{xy} \mid)$. State of an object is represented by a conjunction of literals that appear in stimuli and/or consequences of one or more behaviors $\beta_{ij} \in B$.
- **Behavior chain** - Complex behavior occurs because a number of primitive behaviors (e.g. β_{ij}) operate sequentially and/or concurrently. Here we focus on the temporal sequencing mechanism that gives rise to a complex behavior. A behavior chain C_i is a temporal sequence of behaviors, $\{\beta_{i_1j_1} : \beta_{i_2j_2} : \beta_{i_3j_3} : \ldots : \beta_{i_kj_k}\}$, where $1 \leq i_m \leq n$, $1 \leq j_m \leq b_{i_m}$ and $1 \leq m \leq k$, where $\beta_{i_mj_m} : \beta_{i_{m+1}j_{m+1}}$ is used to denote that these two behaviors are contiguous and occur immediately next to each other in time, with the former behavior preceding the latter (this notation is similar to the notation in our previous work [3]). Such a chain is said to be composable from the behavior space B (denoted by $C \lhd B$) if behaviors in the chain are elements of B.

A complex behavior in which behaviors execute concurrently can be decomposed into individual chains, if there are no interactions between them affecting truths of literals in stimuli and consequences of their behaviors. If β_{11} precedes β_{27} and β_{34}, both of which operate concurrently and β_{47} succeeds these concurrent behaviors, then this is decomposed into two chains, $\{\beta_{11} : \beta_{27} : \beta_{47}\}$ and $\{\beta_{11} : \beta_{34} : \beta_{47}\}$. If this chain is to execute from the initial world state I, it is necessary that $(I \Rightarrow s_{11})$. In the chaining mechanism, the action of an earlier behavior changes the situation in such a way that the newly changed part of the situation in conjunction with the universe U implies stimulus for the next behavior in the chain. Behaviors can be chained together to generate action-streams such as trajectories for a robot, question sequences in coaching, etc. Behavior chains are responsible for fulfilling tasks (defined in the discussion on *system task space* in this section).

• **Complexity of a goal** - A primitive goal g_i to be fulfilled by the multi-agent system is considered to be specified as a 2-tuple $< I_i, F_i >$, where I_i and F_i denote initial and final states of object i respectively, which are assumed to be expressed in conjunctive form (An object is an entity manipulated by one or more agents). Effort required to achieve a goal depends not only on final state but on initial state as well. Hence initial state is included in the definition of a goal here. This definition of goal should not be viewed as more restrictive than the conventional definition (desired final state). If one does not want to separately compute and specify I_i for each goal g_i, one can just replace each I_i by the current world state or some other arbitrary initial state that one prefers to specify.

The complexity of a primitive goal g_i, denoted by $c(g_i)$ is defined as the minimum number of behaviors required to fulfill it, starting from I_i, with multiple occurrences of behaviors counted separately (this is essentially the length of shortest chain that can execute from I_i and lead to F_i). A non-primitive goal consists of a set of such primitive goals. The complexity of a non-primitive goal G, denoted by $C(G)$ is defined as the sum of complexities of the individual primitive goals. The goals in G may be interfering, (if one goal is subgoal of another, fulfilling one may mean fulfilling another), however for the purpose of computing complexity, it is assumed that each goal is fulfilled independent of other goals. When all goals from a set are fulfilled independently from their respective initial states, the minimum number of actions to fulfill all of them will be no less than the sum of complexities of individual goals. Hence the complexity of a goal set is defined as the sum of complexities of individual goals. It does not make sense to take into account lengths of longest chains while computing goal complexities becauses one can construct goal fulfilling chains that are arbitrarily long because of redundant actions (e.g. cycles). A set of goals G' is defined to be more complex than a set G if G' is obtained from G by replacing one or more goals $g_i \in G$ by more complex goals and/or adding more goals to G. $< I_i, F_i >$ notation can represent both goals of achievement and goals of maintenance (e.g. keep the battery charged, for this initial and final values of battery voltage will be the same). The number of maintenance goals is considered to be constant in our analysis.

• **System task space** - We are interested in defining a measure of the number of tasks that are potentially fulfillable by the system. These correspond to all possible temporal chains of behaviors. Where this is achieved by executing behaviors in a temporal sequence, tasks can be enumerated by the total number of chain fragments that are possible. Thus the chain $\{\beta_{11} : \beta_{22} : \beta_{33}\}$ fulfills a task that is different from $\{\beta_{11} : \beta_{22}\}$. The chains executable from a given world state W can be represented by a tree, the root of which is W. Here we are interested in computing the upper bound on the number of tasks fulfillable when the number of occurrences of each behavior is bounded. In the best case, initial state I would imply stimuli of all behaviors of all agents. One can form separate behavior trees with each behavior as their root. Let us assume that a tree can have at most m levels. In the best case (when size of behavior tree of the multi-agent system is maximum), all behaviors of all agents will occur after all nodes (which are behaviors), at all levels in the tree. Since there are $| B |$ trees like this, the total number of chains is

$$| B | \left(1 + \sum_{i=2}^{m} | B |^{i-1} (i-1)\right)$$

This number is defined to be the system task space. Actual number is likely to be less than this since not all behaviors are likely to succeed all nodes due to absence of stimuli resulting from interactions among the chains. There will be a number of chains that are repeated within a tree and among trees, however the chains that precede them are different (since each tree has a different root) and hence they may not fufill the same tasks. To fulfill a goal, one or more tasks have to be fulfilled.

• **Degree of reactivity** - Here we are interested in reactivity along spatial dimension, the amount of internal state rather than the response time. Let the number of literals occurring in stimulus s_{ij} of a behavior β_{ij} whose truth value is decided by the agent i based on external information gathered by sensors be $| s'_{ij} |$, then degree of reactivity of the behavior, r_{ij} is defined as the ratio $\frac{|s'_{ij}|}{|s_{ij}|}, 0 \leq r_{ij} \leq 1$ (where s'_{ij} is derived from s_{ij} by dropping literals whose truth is not determined based on external information gathered by sensors). If $s_{ij} = (a_1 \bigvee a_2) \bigwedge (b_1 \bigvee b_2 \bigvee b_3) \bigwedge (c_1 \bigvee c_2 \bigvee \neg a_1) \bigwedge d_1$ and truth of only a_1, b_1, d_1 is determined based on external information gathered by sensors, $r_{ij} = 3/8$ since $s'_{ij} = a_1 \bigwedge b_1 \bigwedge \neg a_1 \bigwedge d_1$. r_{ij} allows us to capture impact of externalization of state on reactivity. The degree of reactivity of an agent is defined as an average of of the degrees of reactivity of its behaviors. The degree of reactivity (R) of a multi-agent system is defined as an average of the degrees of reactivities of its agents.

• **Degree of deliberation** - The degree of deliberation of a behavior β_{ij} denoted by d_{ij} is defined as $(1 - r_{ij})$. This definition is extended to define the degree of deliberation of an agent and degree of deliberation of a multi-agent system (D) in the same way as the definition of degree of reactivity.

• **Marker** - It is a percept denoted by M_i and is described by a pure conjunction of its features. For example, a colored cube kept on a flat surface can serve as a marker and be described as $cube(x) \bigwedge is\_face\_of(x, y_1) \bigwedge color(y_1, red)$ $\bigwedge is\_face\_of(x, y_2) \bigwedge color(y_2, green)$ $\bigwedge is\_face\_of(x, y_3) \bigwedge color(y_3, yellow)$ $\bigwedge is\_face\_of(x, y_4) \bigwedge color(y_4, orange)$ $\bigwedge is\_face\_of(x, y_5) \bigwedge color(y_5, purple)$. The length of this conjunction (here 11) is the strength of a marker, denoted by $| M_i |$. The strength of a marker captures the extent to which one can exploit the marker in externalizing state.

A marker M_i is at least as strong as a marker M_j if $(M_i \Rightarrow M_j)$, it is stronger if in addition to this, $| M_i | > | M_j |$. The strength of a set of z markers is defined as the sum of their individual strengths,

$$\sum_{i=1}^{z} | M_i |$$

The strength of a set of markers present in an environment is a direct measure of their contribution to complexity of the environment. An environment E' is defined to be more complex than an environment E if E' is obtained from E by adding one or more markers to E or replacing existing markers by stronger ones. The complexity of an environment E is denoted by E_c. If E' is derived from E by adding a set of z markers M_i to E, then,

$$E'_c = E_c + \sum_{i=1}^{z} | M_i | .$$

If z existing markers M_i from E are replaced by M'_i, then

$$E'_c = E_c + \sum_{i=1}^{z} | M'_i | - \sum_{i=1}^{z} | M_i | .$$

The markers need not be physical entities in an environment, e.g. an electronic mail with a certain title can serve as a marker to which softbots react.

The term marker is used in this paper to refer to (a) 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 (we consider these as the primary mechanisms of externalizing internal state), e.g. if a robot is supposed to collect all tennis balls except those near a cupboard, one way to design this behavior is to store 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, if a sensor s_1 is noisy and sensor s_2 delivers more accurate readings, one can create a marker that s_2 can sense and then s_2 can be used instead of s_1. Markers can also be used to reduce perceptual computations but these uses of markers are not considered here.

• **Coupling** - Coupling c_{ijmq} is said to exist between two behaviors β_{ij} and β_{mq} if values of some variables in some literals in s_{mq} are set by conditions of the form $(A \Rightarrow B)$ (explained later here) in s_{ij} or vice versa and is defined as the sum $(k + u)$ where k is the number of literals in s_{mq}, one or more variables of which are assigned values by conditions in s_{ij} and u is the number of literals in the corresponding conditions in s_{ij}, e.g. if an agent is to place painting brushes near any window, then there will be no coupling between behavior that picks up a painting brush (say β_{11}) and the behavior that puts it near a window (say β_{22}), in that case $s_{11} = graspable(x) \bigwedge brush(x)$ and $s_{22} = in\_hand(x) \bigwedge brush(x) \bigwedge agent\_at(y) \bigwedge window(y)$. If it is decided that big brushes should be kept near big windows and small brushes near small windows, the stimuli are modified to be $s_{11} = graspable(x) \bigwedge brush(x) \bigwedge (small(x) \Rightarrow assign(Y, sm)) \bigwedge (big(x) \Rightarrow assign(Y, bg))$ where it is assumed that in evaluating truth of a wff of form $(A \Rightarrow B)$ (where A is expressed in CNF and B is an atomic formula or a conjunction of atomic formulas of arity 2, introduced for the purpose of making assignments), B is evaluated to be true if A is true and $assign(x, a)$ sets value of the variable x to a. The other stimulus s_{22} is modified to be $(in\_hand(x) \bigwedge brush(x) \bigwedge agent\_at(z) \bigwedge window(z) \bigwedge size(z, Y))$. For the purpose of computing u in the coupling, wffs of the form $(A \Rightarrow B)$ are converted to $(\neg A \bigvee B)$. Here β_{11} and β_{22} are behaviors that are coupled. The clauses involved in coupling here are $(small(x) \Rightarrow assign(Y, sm)), (big(x) \Rightarrow assign(Y, bg))$ and $size(z, Y)$. Hence the actual coupling c_{1122} is 5 (2 + 2 + 1). The coupling of an agent is computed as sum of the couplings between pairs of its behaviors and the couplings between its behaviors and behaviors of other agents. The coupling of a multi-agent system is computed as the sum of the couplings between distinct behavior pairs from B. Clearly, coupling between behaviors of different agents makes agents less autonomous and requires explicit parameter passing. Coupling is different from the causal connectivity between the neighbors of a behavior chain (by causal connectivity between neighbors, we mean consequence c_{ij} of a behavior β_{ij} providing literals needed to be true for stimulus s_{gh} of next behavior β_{gh} in same behavior chain to be true. In such a case, $\beta_{ij} : \beta_{gh}$ will be a part of that chain). The coupling defined here captures dependency among stimuli of behaviors that is forced to occur to fulfill specified goal. The stimuli are assumed to be modified by a human to handle the dependency (an agent is not considered to be autonomous enough to modify the stimuli of its behaviors on its own). The coupling can be used to define autonomy of an agent and a multi-agent system. The higher the coupling, the lower is the autonomy and modularity.

3 Results

In this section, we discuss our results on interactions between complexity of goals, coupling and degree of reactivity of a multi-agent system.

Theorem 1. As the complexity of a goal set increases, the coupling of the multi-agent system changes from C to C' where $C' \geq C$, when $\mid B \mid$ is kept constant.

Proof - Let the goal set G' be derived from the goal G by adding m tuples of the form $< I_i, F_i >$ to G (so that

$$C(G') = C(G) + \sum_{i=1}^{m} c(g_i),$$

leading to $C(G') > C(G)$) or replacing existing tuples by more complex goals $(C(g_i') > C(g_i))$, so that G' is more complex than G. These tuples specify goals of achievement since goals of maintenance are assumed to be invariant. 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 stimuli and/or consequences of behaviors so that existing behaviors are chained in a certain way to fulfill the more complex goal set. We consider how the stimuli of behaviors in the current multi-agent system can be modified to fulfill all 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 stimuli of two behaviors β_{ij} and β_{mq} where s_{ij} contains conditions that assign values to variables in literals in s_{mq}. One can modify the values that are being assigned or negate predicates making assignments (e.g. $assign(x, a)$) 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_{ij} 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_{ij} can be changed to include $(small(x) \Rightarrow assign(Y, sm)) \bigwedge \neg small(x)$ or $(small(x) \Rightarrow assign(Y, nil))$. However these changes leave k, u (as defined in the discussion on coupling in previous section) unchanged, this argument can be repeated for other pairs of stimuli. Hence coupling of an agent and the system does not change. Hence $C' = C$.

Case 2. Stimuli are changed by adding new conditions to s_{ij} and/or increasing the number of literals and/or variables in s_{mq} that 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 $(k + u + k' + u') > (k + u)$. 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 coupling of at least one behavior pair increases, $C' > C$, hence the proof. □

Not having coupling between agents will make them more autonomous but will prevent some concurrent behaviors from occurring. Keeping the coupling within an agent in fact introduces some sequentiality, e.g. in the example of moving brushes discussed above, same agent is responsible for moving and dropping the brush at desired location, once it picks up the brush. 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 $(small\,(x) \bigwedge brush\,(x))$, $(big\,(x) \bigwedge brush\,(x))$ and $(medium\,(x) \bigwedge brush\,(x))$ and corresponding behaviors for dropping the brushes. This suggests another dimension of analysis.

Theorem 2. If coupling of multi-agent system is changed from C to C', $C' < C$, then fulfilling original goals requires either B to be modified to B' such that $\mid B' \mid > \mid B \mid$ or introduction of m markers, $m \geq 1$.

Proof - Since coupling of the system is reduced, there exists a pair of behaviors, β_{ij}, β_{mq}, such that the coupling between the behaviors is reduced. This means that either the number of literals in s_{mq}, variables of which were assigned values by conditions in s_{ij} were reduced and/or the number of conditions in s_{ij} making truth assignments and/or the number of assignment predicates in s_{ij} were reduced. These changes will either result in variables in literals in s_{mq} not having any values assigned to them or variables that have some fixed values assigned arbitrarily(these values are not set by conditions in s_{ij}, e.g. instead of letting a condition in s_{ij} set value of Y in $at(Y)$ in s_{iq}, one can force it to some arbitrary value $at(5)$ and reduce the coupling) or variables that have incorrect values assigned, resulting in fewer goals fulfilled or an undesired behavior. In such a case, one will have to add behaviors to B to fulfill the unfulfilled goals, leading to $\mid B' \mid > \mid B \mid$, (as discussed above in the case of moving brushes of different sizes) or 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 $s_{11} = graspable(x) \bigwedge brush(x) \bigwedge (small(x) \Rightarrow assign(Y, sm)) \bigwedge (big(x) \Rightarrow assign(Y, bg))$, $s_{22} = in\_hand(x) \bigwedge brush(x) \bigwedge agent\_at(y) \bigwedge window(y) \bigwedge size(y, Y)$ to $s_{11} = graspable(x) \bigwedge brush(x) \bigwedge has\_word(x)$ ($has\_word(x)$ means that x has a word written on it) and $s_{22} = in\_hand(x) \bigwedge brush(x) \bigwedge agent\_at(y) \bigwedge window(y) \bigwedge on(x,z) \bigwedge word(z) \bigwedge size(y,z))$, but this introduces $m > 0$ markers (This also makes the environment of the multi-agent system more complex), hence the proof. □

Lemma 1. Adding a marker increases degree of reactivity of a multi-agent system.

Proof - Since markers are added so that truth of one or more literals in one or more stimuli can be decided purely based on external information gathered by sensors, when a marker M_x is added, $\exists \beta_{ij}, l(in(s_{ij}, l) \bigwedge \neg in(s'_{ij}, l) \bigwedge (M_x \Rightarrow l))$, which means that there exists a behavior such that some literal from its stimulus whose truth was not decided based on purely external information can now be assigned truth value purely based on the external information available in the form of marker that can be sensed. In that case, r_{ij} increases by at least $\frac{1}{|s_{ij}|}$. Hence R increases. Hence the proof. □

When we transfer an internal state to external, we are making that state publicly available and that may be perceived by other agents. Hence evaluating the effect of adding a marker on the task fulfilling capabilities of the system is important. We now explore the relation between length of stimulus of a behavior and the change in environmental complexity that occurs when we try to make a behavior more reactive.

Theorem 3. Given two behaviors of same degree of reactivity, the one with longer stimulus makes an environment more complex than the one with a weaker stimulus, when their degrees of reactivity are increased to unity by introducing markers of unit strength.

Proof - Let us consider two behaviors β_{ij} and β_{pq}, such that s_{ij} is longer than s_{pq}. Hence $\mid s_{ij} \mid > \mid s_{pq} \mid$. If the degrees of reactivity of the two behaviors are same $(r_{ij} = r_{pq})$, $\frac{|s'_{ij}|}{|s_{ij}|} = \frac{|s'_{pq}|}{|s_{pq}|}$ Using these two relations, it can be inferred that $(\mid s_{ij} \mid - \mid s'_{ij} \mid) > (\mid s_{pq} \mid - \mid s'_{pq} \mid)$, which means that the number of markers m' required to raise r_{ij} to unity is more than the number of markers m'' required to raise r_{pq} to unity, since all markers have unit strength. When an environment E is changed to E' to raise r_{ij} to one, $E'_c = E_c + m'$. When an environment E is changed to E'' to raise r_{pq} to one, $E''_c = E_c + m''$. Since $m' > m''$, $E'_c > E''_c$. Hence β_{ij} makes an environment more complex than β_{pq}. □

4 Constraint-based specification

The conclusions here have strong implications for all kinds of agents, e.g. some search algorithms of a softbot trying to retrieve data fulfilling certain characteristics from a database, can be considered as its sensors and markers can be introduced in the database to assist the search algorithms in performing better. We consider an agent system to be a ***situated software*** and propose that realistically, agents should be designed using a specification of requirements, just like the conventional software. Below we identify various constraints involved in designing a multi-agent reactive system.

1. $\{r_{ij} \in [l_{ij}, h_{ij}], 1 \leq i \leq n, 1 \leq j \leq b_i\}$. Such constraints specify that degree of reactivity of a behavior should fall within a certain range, l_{ij}, h_{ij} being the lower and upper bounds. In practice, certain features can be sensed faster, then reactive behaviors are faster than deliberative. But too much reactivity can lead to myopic behavior or infinite cycles. Hence such constraints are needed.

2. $\{b_i \in [l_i, h_i], 1 \leq i \leq n\}$. These constraints limit the

number of behaviors of agents to certain ranges.
3.

$$\sum_{i=1}^{n} b_i \in [l_{|B|}, h_{|B|}]$$

which limits the size of behavior space of the system.
4.$\{c_{ijiq} \in [l_{ijiq}, h_{ijiq}]\}$ which limit coupling between behaviors of an agent where $1 \leq j \leq b_i$ and $1 \leq q \leq b_i, j \neq q$.

$$\{\sum_{i=k}^{k} c_{ijmq} \in [l_{ic}, h_{ic}], 1 \leq k \leq n\}$$

limits coupling of an agent.

$$\sum_{i=1}^{n}\sum_{p=i}^{i} c_{pjmq} \in [L_c, H_c]$$

limits total coupling of the system. The coupling is a measure of independence within an agent's behavior library. Higher the coupling, lower is the independence. This independence is related to autonomy of an agent. Additional constraints expressing desired autonomy can be developed.
5.

$$\forall g_i(g_i \in G \Rightarrow fulfills(B, g_i)), \sum_{i=1}^{|G|} c(g_i) \in [l_G, h_G]$$

which expresses that the system should have goal fulfilling capability within a certain range.
6. Let the complexity of an environment E be denoted by E_c. A constraint may be imposed on the environmental complexity through $E_c \in [l_E, h_E]$. This limits the number of markers and hence the amount of externalization of state.
7. Constraints will have to be specified for the fulfillment of individual goals. Such a constraint will specify that a chain of behaviors fulfilling that goal should exist. $\exists C_i((C_i \lhd B) \bigwedge(fulfills(C_i, g)))$ specifies that there should exist a chain C_i composable from B that fulfills goal g.

This set of constraints leads to a constraint-based specification of a multi-agent system. The set of constraints mentioned above can be encoded into propositional logic and the problem of designing an agent system can be cast as a propositional satisfiability problem. In particular, the problem of finding a model for the encoding will correspond to the problem of designing a multi-agent reactive system to satisfy the specified constraints. Using this framework to establish the bounds on the size of the encoding and exploring the utility of other types of encodings is our future work. Though this may look impractical in the light of complexity of real world, agents designed in this way will provide an approximation to agents that work in the real world. Such approximations can be iteratively refined by tuning some constraints, e.g. adding new constraints to account for noise in sensors (sonars of a robot, *finger* command of a softbot) and problems in effectors (electric motors of robots, *ftp* command of softbots).

5 Conclusion

It has argued that a behavior many times looks complex because of the complexity of environment in which it occurs [5]. However if we transfer internal state to external world by creating markers, it makes the the environment more complex. We have presented only those constraints that are independent of architecture and representation. Even if one probes into hybrid architectures that have a planner and a reactor or planner, sequencer and a reactor and use representations like fuzzy logic, potential fields and finite state automata, our constraints continue to be relevant. Our results and formalism can be used to derive a number of other specific tradeoffs, when specific information about agents, environments and goals is available. Such specific information will provide additional constraints. Constraints prohibiting occurrence of an undesirable behavior can be added to the constraints we outlined here. An example of a system that could be designed in this way is a set of software agents that have behaviors for gathering information, sending and forwarding e-mails and outlining schedules for meetings. In bottom up approach, it is not clear what set of behaviors to start with and the scalability of this approach is questionable. Top down decomposition-based approaches require knowledge in the form of reduction schemas, and this is rarely completely available. Our approach provides a more homogeneous framework in which agents are viewed as models of a set of constraints.

References

[1] **Brooks R. A.**, Intelligence without representation, Artificial intelligence 47, 1991, 139-159.

[2] **Grosz, Barbara J, and Davis, Randall**, AAAI Report to ARPA on 21st century intelligent systems, AI Magazine, Fall 1994, 10-20

[3] **Amol D. Mali, and Amitabha Mukerjee**, Robot behavior conflicts: Can intelligence be modularized ? Proceedings of AAAI-94, 1279-1284.

[4] **Stanley J. Rosenschein and Leslie Kaelbling**, A situated view of representation and control, Artificial Intelligence 73, 1995, 149-173.

[5] **Simon H.A.** The sciences of the artificial, The MIT Press, 1969.