A New Control Architecture Combining Reactivity, Planning, Deliberation and Motivation for Situated Autonomous Agent

François Michaud, Gérard Lachiver and Chon Tam Le Dinh
Department of Electrical and Computer Engineering
Université de Sherbrooke
Sherbrooke (Québec) CANADA J1K 2R1
{michaudf, lachiver, ledinh}@gel.usherb.ca

Abstract

Intelligent behavior can be observed from both natural and artificial systems, but still the notion of intelligence is very difficult to define. We propose a new control architecture allowing the combination of reactivity, planning, deliberation and motives for building intelligent agents that must deal with real or other kinds of environments suited to their life purposes. This control architecture tries to unify the principles and characteristics associated with intelligence. It uses behaviors as its basic control components. These behaviors are selected dynamically and their actions are combined according to the intentions of the agent. Introspection of its reactions and its knowledge is one major new ability given to the agent by this architecture. This way, the agent is not only able to adapt to the environment, but also to its own capacities. One implementation for experimenting with a simulated environment for mobile robots is presented here to illustrate the use of this architecture.

1. Introduction

Intelligence studied in its various aspects is a source of inspiration in Artificial Intelligence (AI). Insights about intelligence are taken from psychology [Simo95], from neuro-ethology [Beer90], from personal intuition and from observation of our own intelligent behavior. In spite of the significant progress made in AI, intelligence is still a difficult notion to define and to entirely reproduce in artificial systems. The fields of research in AI follow particular guidelines about the principles associated with intelligence. For example, Saridis [Sari83] believes that intelligence is based on a three-level hierarchy (execution, coordination and organization), layered according to the decreasing precision with increasing intelligence principle. Albus [Albu91] proposes also a hierarchical architecture but with a central world model and according to a functional decomposition of intelligence. In contrast, multi-agent research [Wern92] believes in distributing intelligence in different knowledge sources that have to work together to solve a task. The behavioral decomposition of intelligence proposed by Brooks [Broo86] also proves to be a very useful approach. Finally, hybrid approaches try to combine the advantages of reactivity and planning by following the previous guidelines [Firb89, Nore95, Simm90, Haye95, Kael86, Donn94].

All of these principles are associated with intelligence, and a way to combine them into a general architecture is needed. A more general architecture must allow the combination of characteristics associated with intelligence like reactivity, emergent functionality, modeling, planning, deliberation, learning, goal, motivation and emotion. But these characteristics are not sufficient nor essential conditions to characterize or to reproduce intelligent behavior into an artificial system. Intelligent behavior must be established based on the ability of the agent to adapt to two things: the environment it is in; and its own capacities or limitations (of its sensing and actuating abilities, its processing and memorizing abilities, and its decision abilities) affecting its ability to interact with the environment.

This paper presents a new control architecture that tries to take into consideration all of these aspects and to combine them while preserving their underlying principles [Mich96]. Section 2 presents the architecture and its characteristics. Section 3 describes the use of this architecture and the mechanisms developed for experimentation using a simulated world for mobile robots. Section 4 shows some results obtained with the simulated environment to illustrate the use of the architecture. Finally, conclusions and future work are outlined.

2. Intentional Selection of Behaviors

This new architecture presents interesting characteristics for building intelligent agents that must deal with real or other kinds of environments suited to their life purposes. It consists of six modules as illustrated in Figure 1. The *Behavioral* module is a behavior-based system [Broo86, Mata92]. It is made of different behaviors connecting sensory information to actuation. It defines the agent's skills for reacting to the situations encountered in the environment. These behaviors all run in parallel and their resulting commands are blended according to their respective importance, to obtain the control actions. The relevance in using each behavior is determined by three recommendation modules. The *External Situation* module evaluates special external conditions in the environment that can affect behavior selection. The *Needs* module selects behaviors according to the needs and goals of the agent. The third recommendation module, called the *Cognition* module, is for cognitive recommendation. This module learns things about the external environment and how the agent operates in it by observing its reactions, behavior selections and from information sensed from the environment. It can then exploit this acquired knowledge or some innate knowledge to plan or to prepare the use of behaviors. Cognitive recommendations can be influenced by behaviors via the *Internal Parameters* link, like they can influence behavior reactivity using the same link. These three recommendation modules suggest the use of different behaviors to the *Final Selection* module, which combines them appropriately to establish the activation of the behaviors (*Behavior Activation*). An active behavior is allowed to participate to the control of the agent, and it is said to be exploited if it is used to control the agent (by reacting to the sensations associated with the purpose of the behavior). Finally, the *Motives* module is composed of motives used to examine and to coordinate the proper working of the other modules. Motives are influenced by the environment, the internal drives of the agent, its knowledge, and by observing the effective use of the behaviors (*Behavior Exploitation*). The agent is then able to adapt its emerging behavior to its needs, its knowledge and its ability to satisfy its intentions.

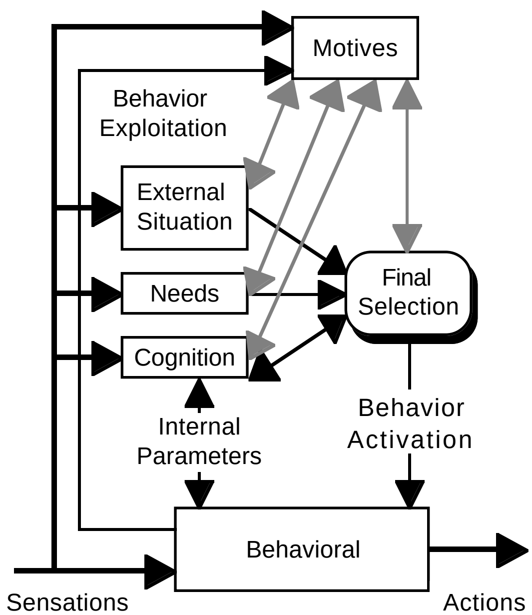


Figure 1: Control architecture proposed

This is a very brief explanation of the modules. However, a better description can be made using an example. The next section describes the mechanisms implemented for the experiments using a simulated world for mobile robots.

3. Module Implementation for the Simulated World

To validate the control architecture proposed, we wanted to develop an autonomous agent having to deal with various goals like managing its energy, its purposes and its well being. To do this, we have used a simulated world for mobile robots called *BugWorld* [Alm93]. An agent in *BugWorld* has a circular body equipped with distance sensors (similar to range finders) for detecting obstacles and targets. For our experiments, the agent is placed in a room where it can find targets and a charging station. The agent must be able to efficiently reach the targets and must survive by recharging itself when needed. The agent knows nothing about the environment it is in, but it has a limited memory to acquire knowledge that can be helpful in its task. For sensing, the agent has at its disposal eight proximity sensors for obstacle, each separated by 45˚ starting from its nose. There are also two target sensors, one on each side. One target in the room is used as a charging station. The agent can also read the amount of energy available, its speed and its rotation (the rotation is only used to indicate when the agent is moving or not). For actions, the behaviors can affect the speed, the rotation or a variable for the color of the agent.

To achieve this task, the agent is going to follow a scenario guiding its general behavior according to what it experiences in the environment. First, the agent starts to acquire knowledge about the environment by following boundaries, reaching a target or a charging station deliberately or not. When the agent is able to recognize its location in the environment, it can start exploring other regions. Eventually, when the agent judges that it knows enough about the environment, it can use this knowledge for reaching memorized targets.

Different AI methodologies can be used with the modules of our control architecture. For the experimentation presented here, the implementation of the modules are inspired by three techniques from other approaches. Fuzzy logic is used for behaviors and for the blending of their control actions, like in Saffiotti *et al.* [Saff93]. It is also used by the *External Situation* module and the *Needs* module for recommending behaviors, and by the *Final Selection* module for combining these recommendations. A topological graph, having some similarities with the work of Mataric [Mata92], is used by the *Cognition* module to construct an internal representation of the environment based on the experiences of the agent. Finally, activation levels as in Maes [Maes91] are used for motives. All of these techniques are described in the next subsections.

3.1. Fuzzy Behaviors

A fuzzy behavior uses rules and linguistic variables to establish the relation between sensations and actions. The processing steps are similar to the ones for fuzzy systems [Lee90], which are fuzzification, rule inference and defuzzification. The only difference here is that rule firing strength is affected by μ_{act}, the activation of the behaviors, given by the *Final Selection* module. The processing steps for the *Fuzzy Behavioral* module are summarized below:

- Fuzzification (1). This operation converts input data into linguistic values (A_i) characterized by a label and a membership value. Figure 2 gives an example of membership functions used for fuzzification of the front sensor of the agent (i.e. its nose). In this case, the fuzzy representation of a sensory input of 15 is given by two linguistic values: a membership of 0.67 for the *Danger-in-front* fuzzy set, and a membership of 0.33 for the *Near-front* fuzzy set.

$$Sensations \rightarrow \mu_{A_i}(Sensation) \tag{1}$$

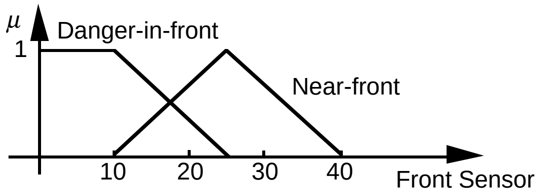


Figure 2: Example of membership functions

- Fuzzy implication of rule *r* for the behavior *j* (2). The operator ⊗ is the minimum and is used for the fuzzy conjunction of the *n* antecedents of rule *r*. The result is called the firing strength of the rule, which is a measure of the contribution of the rule to the fuzzy control action [Lee90]. The firing strength of the rule is associated with its fuzzy consequence, which is a linguistic variable (*B* or *C*) for a fuzzy control action. This processing step is repeated for all the rules of a behavior, and all the activated behaviors. When NOT is used in front of an antecedent, the complement [Lee90] of its membership value is used. Examples of rules are presented in Figure 3 for the EMERGENCY behavior, and in Figure 4 for the TURN180 behavior. For the rule *Slow-down-danger* of Figure 3, if *Danger-in-front* has a membership value of 0.67 and *Speed-null* has a membership value of 0 (which indicates that the agent is moving), then the membership value of *Slow-down-fast* is 0.67.

$$\mu_{B_{rj}}(Action) = \otimes\left[\mu_{A_n}(Sensation)\right] \tag{2}$$

- Adjustment of the firing strength of the rules according to the activation μ_{act} of the behavior *j* (3). The minimum is also the fuzzy conjunction operator used here. For example, if the activation of the EMERGENCY behavior is 0.5 and the firing strength of rule *Slow-down-danger* is 0.67, then the adjusted membership value of the fuzzy consequence *Slow-down-fast* for this rule is 0.5.

$$\mu_{C_{rj}}(Action) = \otimes\left[\mu_{B_{rj}}(Action), \mu_{act}(j)\right] \tag{3}$$

- Union of the fuzzy consequences (4). The fuzzy disjunction operator ⊕ maximum is used to combine membership values for identical fuzzy consequence. For instance, the rule *Danger-in-front* of the EMERGENCY behavior and the rule *Immobilization* of the TURN180 behavior have the same fuzzy consequence. If their respective adjusted membership values for *Slow-down-fast* are 0.5 and 0.9, then their union gives a membership value of 0.9.

$$\mu_{C_o}(Action) = \oplus\left[\mu_{C_{rj}}(Action)\right] \tag{4}$$

- Defuzzification using the center of area method (5). This step converts the fuzzy consequences into a 'crisp' (numerically precise) output. The parameter w_{Cx} is the support value at which the membership function for C_x reaches the maximum value μ_{Cx} (the average is used when there is two support values with the same membership strength), and *x* represents the linguistic variables for a common control action. This step allows the smooth blending of the fuzzy commands given by the activated and exploited behaviors. For example, the *Slow-down-fast* fuzzy consequence is a linguistic variable for the *Acceleration* control action (associated with speed control). Its *w* parameter is -3.25 for a membership value of 0.5. For the same control action, if there is another fuzzy consequence called *Accelerate* with a membership value of 0.1 and *w* = 4, then the resulting control action *Acceleration* is -2.04.

$$Action = \frac{\sum_x \mu_{C_x}(Action) \cdot w_{C_x}}{\sum_x \mu_{C_x}(Action)} \tag{5}$$

Twelve behaviors are used in our experiments. The first is EMERGENCY, which is responsible for moving the agent when immediate danger is detected in its front. The rules are presented in Figure 3. Using this behavior, the agent slows down if it is in front and very close to an obstacle; it turns away from an obstacle at its side (the variable *x* is for *left* or *right*, and *y* denotes the opposite direction); and it makes a wide turn left when the obstacle is right in its front.

<Slow-down-danger>
IF Danger-in-front
AND NOT (Speed-Null)
THEN Slow-down-fast
<Danger-*x*>
IF Danger-front-*x*
AND NOT (Danger-front-*y*)
THEN Turn-*y*
<Danger-in-front>
IF Speed-Null
AND Danger-in-front
AND Danger-front-right
AND Danger-front-left
THEN Turn-left-big

Figure 3: Rules for the EMERGENCY behavior

Other behaviors are AVOID to move away from obstacles, SPEED to maintain a constant cruising velocity, ALIGN to follow boundaries, TARGET to search for a target, RECHARGE to search for a charging station and to energize the agent, BACKING to move back, MADNESS to make the agent turn around on itself, TURN90 to move away from a boundary, TURN180 to make a U-turn, ALARM to express some internal state of the agent by changing its color to red, and a behavior for identification of topological states (used by the *Cognition* module: see Section 3.4). Rules for the TURN180 behavior are presented in Figure 4. The behavior starts by slowing down the agent. It then makes it turn away from a boundary until the other side is perceived by the agent. The underlined antecedents and consequences of the rules are adjusted by the *Cognition* module via the *Internal Parameters* link before using the behavior, to select the rotation side.

```
<Immobilization>
    IF      NOT(Speed-almost-null)
    THEN    Slow-down-fast
<Turn-180-left>
    IF      NOT(Side-left-near)
    AND     Speed-almost-null
    THEN    Turn-left
<Turn-180-left-end>
    IF      Side-left-near
    AND     Back-near
    AND     Speed-almost-null
    THEN    Turn-left
```

Figure 4: Rules for the TURN180 behavior

3.2. *Motives*

The agent has five basic goals in the environment: to find a charging station and recharge itself; to reach targets; to detect improper use of its behavior according to its intentions; to explore the environment and acquire knowledge from it; and to use this knowledge when the agent judges it is accurate enough. Motives are responsible for coordinating and supervising these goals according to the actual experiences of the agent in the environment. To do so, ten motives are used by the agent. Each one has its own activation mechanism and can be influenced by sensations (internal or external), behavior activation or exploitation, internal variables of the recommendation modules, or by other motives. Four groups of motives are used:

- Physiological motives, like HUNGRY and EAT, are used to monitor the energy level of the agent and to control the use of the RECHARGE behavior via the *Needs* recommendation module. The agent shows opportunism by wanting to recharge when it reaches a charging station, even if it is not hungry;

- Good Operation motives. These motives are particularly influenced by the *Behavior Exploitation* link to detect improper use of behaviors. Because behaviors are fuzzy, *Behavior Exploitation* is a fuzzy measure defined in relation (6), approximating the contribution or the importance of behavior *j* to the fuzzy control actions formulated before defuzzification. It combines the activation of a behavior with its reactivity to the environment.

$$\mu_{exp}(j) = \mu_{act}(j) \otimes \left(\oplus \left[\mu_{B_{rj}}(Action) \right] \right) \qquad (6)$$

  Two motives are in this group. The motive DISTRESS is used to monitor the proper working of behaviors like EMERGENCY, AVOID and SPEED. These first two behaviors must normally be exploited very briefly to move the agent away from trouble areas. However, if their μ_{exp} remains approximately constant for a long period of time, this may be a sign of conflict between the behaviors used. For the SPEED behavior, a full exploitation for a long period of time is also a sign of trouble indicating that the agent is not able to reach its desired velocity. The motive DECEPTION is the other motive in this group. This motive increases when the agent is moving away from a target or a charging station, detected by a decrease in the exploitation of the TARGET or RECHARGE behaviors respectively. This motive influences the use of the TURN180 behavior via the *Cognition* module;

- Accomplishment motives. Only one motive, called FULFILLMENT, is used in this group to monitor when the agent needs to find targets;

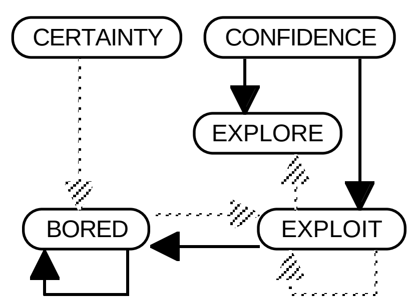


Figure 5: Cognition motives

- Cognition motives supervise the acquisition and the use of the knowledge in the *Cognition* module. These motives are illustrated in Figure 5. A solid arrow indicates a positive influence from another motive, as opposed to a shaded one. The motive CONFIDENCE is associated with the agent's ability to locate itself in previously memorized topological sites. When the agent is able of doing so, the motives EXPLORE and EXPLOIT are excited. The motive EXPLORE directly uses the level of excitation of CONFIDENCE to influence the cognitive recommendation of the TURN90 behavior. The motive EXPLOIT increases gradually when CONFIDENCE is greater than zero, indicating the increasing ability of the agent to know where it is in the environment. EXPLOIT is also excited when the topological graph is full. When EXPLOIT is sufficiently excited, it inhibits the motive EXPLORE to stop the acquisition of knowledge and to use the topological graph constructed for going toward memorized targets. The ability to plan a path using the topological graph is reflected by the CERTAINTY motive. During the exploitation of the topological graph, the BORED motive increases when no path is planned (toward some target that has not already been visited during the exploitation of the topological graph) or when no target is reached. Eventually, the motive BORED is fully excited and reinitializes the EXPLOIT motive. Exploration is then resumed.

3.3. *External Situation and Needs*

These two modules recommend the use or the inhibition of behaviors. They are implemented using fuzzy logic. The operations are similar to those presented in relations (2) and (4), except that the results are fuzzy measures of the desirability or the undesirability of behaviors. The difference between these two modules is that the *Needs* module can use motives as antecedents in its rules. Figure 6 shows the rules used by the *External Situation* module, and Figure 7 presents rules for the *Needs* module. In these rules, an undesired behavior is a consequence preceded by NOT.

```
<Danger>
    IF      Danger-in-front
    OR      Danger-front-right
    OR      Danger-front-left
    THEN    EMERGENCY
<Obstacle>
    IF      Obstacle-in-front
    THEN    AVOID AND NOT(TARGET)
<Normal>
    IF      NOT(Obstacle-in-front)
    THEN    SPEED AND ALIGN
<Topological states>
    IF      NOT(Speed-null)
    AND     NOT(Rotation-null)
    THEN    TOPOLOGICAL STATE IDENT.
<Charging>
    IF      Speed-almost-null
    AND     Charging-station-visible-left
    AND     Charging-station-visible-right
    THEN    NOT(ALIGN)
```

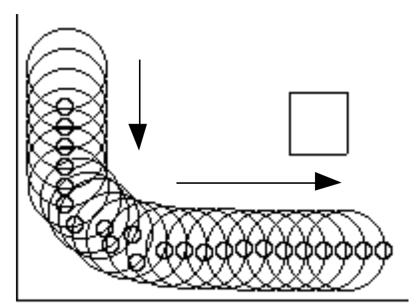
Figure 6: Rules for the *External Situation* module

```
<Want-to-recharge>
    IF      Want-Recharge
    THEN    RECHARGE, NOT(MADNESS)
            NOT(TURN90) AND NOT(TARGET)
<Charging-station-near-x>
    IF      Want-Recharge
    AND     Charging-station-visible-x
    THEN    NOT(SPEED)
<Charging-station-nearer-x>
    IF      Want-Recharge
    AND     Charging-station-nearer-x
    THEN    NOT(ALIGN)
<Difficulties>
    IF      Distress-exists
    THEN    BACKING, ALARM
    AND     NOT(ALIGN)
<Accomplishment>
    IF      Fulfillment-small
    THEN    TARGET
<Happiness>
    IF      Fulfillment-big
    THEN    MADNESS, NOT(SPEED)
    AND     NOT(ALIGN)
```

Figure 7: Rules for the *Needs* module

3.4. *Topological Graph*

Knowledge, acquired by the agent from its experiences in the environment, is represented using a topological graph. The graph is constructed from topological states identified by the TOPOLOGICAL STATE IDENTIFICATION behavior. This behavior examines sensations coming from the front, the back and the two sides of the agent. The presence or the absence of obstacles at a certain distance in these four directions is used to infer one of 16 possible topological states. Figure 8 gives an example of topological states identified as the agent turns a corner and passes near an obstacle while following boundaries.



Right side - Right side - Right side - Right side - Right side -
Right corner to turn - Right corner to turn -
Dead-end right - Dead-end right - Right corner turned -
Right corner to turn - Right corner turned -
Right corner turned - Right side - Right side - Right side -
Right side - Right side - Right side - Right side -
Corridor - Corridor -Corridor

Figure 8: Example of identification of topological states

Two types of nodes are constructed using these topological states according to their characteristics: identification of the same topological state like *Right side*, *Left side*, *Corridor* or *Nothing*, for a consecutive number of cycles, is associated with a stable landmark node; other topological states identified between two landmark nodes are used to construct a transition node. For a transition node, the topological state sequence is analyzed to characterize and to approximate the rotation made by the agent. This is done using regular expressions for a lexical analysis [Aho88] of the topological states identified during this sequence. Figure 9 presents some of the regular expressions used. On the right side of the expressions, the operators indicate the number of the same consecutive topological state that must be in the sequence: '+' indicates more than one; '-' indicates only one; '?' is for zero or one. These expressions are evaluated in parallel after the consecutive identification of the same topological state. If a topological state does not correspond with the active state in a regular expression, then the regular expression is dismissed. Priority is assigned according to the order of definition of the regular topological expressions (the first ones have priority on the others). When an expression is completely validated, the result at the left side of the expression is memorized. If there is no result obtained for the sequence of topological states identified during a transition, the best guess, according to the expressions evaluated, is used. These regular expressions can be seen as symbolic behaviors reacting to topological state sequences to characterize the transition made by the agent.

Internal *Right* Corner - 110° =
Right corner to turn+ Dead-end *right*+
Right corner turned- *Right* corner turned-
Right corner turned-
Internal *Right* Corner - 90° =
Right corner to turn+ Dead-end *right*+
Right corner turned- *Right* corner turned?
External *Right* Corner - 110° =
Nothing+ *Right* side+ Nothing+
External *Right* Corner - 90° =
Nothing+ *Right* side - Nothing -
U Turn *Right* - 180° =
Right corner turned- Against a wall-
Left corner turned-

Figure 9: Example of regular topological expressions

Nodes are constructed as the agent moves in the environment. They memorize the landmark type (the topological state for a stable landmark, or the result obtained from a transition landmark analysis), its length, the orientation of the agent (for stable landmark only, cumulated from previous nodes according to the rotation approximated by transition nodes) and the number of the branch in construction in the graph. Other information like the number of visits to the node, the presence of a goal like a target or a charging station, the occurrence of a motive like DISTRESS or DECEPTION, the use of particular behaviors like TARGET and MADNESS, and path planning variables are other fields accessible in a node. Nodes are connected together with bidirectional links. These links have information about the anticipation of the state of the connected node, making the graph reversible. Uncertainty measure concerning the length of the node, which occurs between a landmark node and a transition node, is also memorized in the link. Figure 10 presents the graph constructed based on the topological states identified in Figure 8.

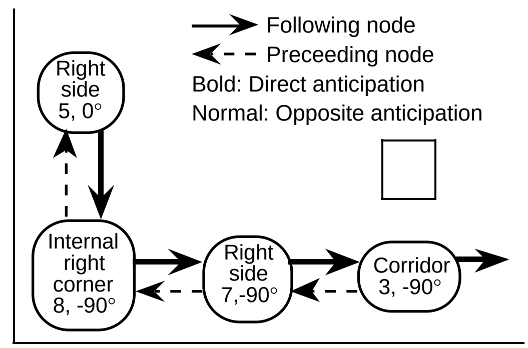


Figure 10: Topological graph

To find out if the agent is located on a previously visited landmark, a search in the constructed nodes must be initiated. During this search, the *Cognition* module tries to find a sequence of three nodes similar to the three most recent nodes constructed. Similarities are evaluated according to the landmark type, the anticipation of the link followed to establish the sequence, the length of the nodes, and sometimes the orientation. When only one similar sequence is detected, the recent nodes can be eliminated and a loop can be established in the topological graph. The agent is then situated in its memorized topological representation and can compare its state according to the following nodes in the graph. If a divergence is observed, a new branch is initiated and the search process is reactivated. In addition, when the graph is full or when the agent wants to exploit it, three buffer nodes are used to locate, if possible, the agent in its topological graph.

Path planning with the topological graph is done by activation spreading from the current location to a node referring to a particular goal. Activation is spread following the links to the nodes, preferring the paths in the same direction of the agent, with the fewest nodes, the smallest length, and avoiding special conditions like DISTRESS, DECEPTION, MADNESS and U-turns. Path planning is also used to optimize the topological graph when the agent wants to exploit it. Only the useful nodes are then kept in the graph. A node is considered to be useful when it has been visited more than once, when it indicates the start of an exploration path, or when it is part of the optimal path from a node referring to a charging station to a node referring to a target.

Cognitive recommendations are binary (which is a special case of fuzzy membership strengths of 0 or 1). They are responsible for making the agent explore the environment by using the TURN90 behavior, and also for making a U-turn by using the TURN180 behavior. The SPEED and ALIGN behaviors are then inhibited to let these behaviors control the actions of the agent. Cognitive recommendations are also responsible for the use of special behaviors when a memorized topological path must be reproduced. Finally, they can also activate the ALARM behavior to transmit an S.O.S. when the agent thinks it does not have enough energy to reach a charging station (based on what it can anticipate from its topological graph).

3.5. Final Selection

The fact of using desirability and undesirability for recommending behaviors has been inspired by the hedonic axiom which indicates that the organisms direct their behaviors to minimize aversions and maximize desirable outcomes [Beck83]. Here, *Behavior Activation* is evaluated based on a hedonic continuum established from the fuzzy desirability and undesirability measures. First, these measures are respectively combined for each behavior using the fuzzy disjunction operator maximum. Then, the desirability is subtracted to the undesirability measure and the behavior is activated if the result is greater than zero. Relation (7) shows these operations where *m* represents the recommendations from the three recommendation modules. So, to be activated, the desirability of the behavior must be higher than its undesirability.

$$\mu_{act}(j) = \max\left(0, \oplus\left[\mu_{des_m}(j)\right] - \oplus\left[\mu_{und_m}(j)\right]\right) \tag{7}$$

4. Experimental Results

For our experimentation, the agent has enough energy for 250 cycles. When it reaches a charging station, the agent detects that it is recharging by sensing an increase of its energy level. It stops recharging when its energy level is maximum. Also, a target reached is inhibited for 200 cycles. The results presented here are parts of longer trajectories followed by the agent. We only want to show the use of the mechanisms described in Section 3 and to illustrate the general behavior of the agent in two different environments.

Figure 11 illustrates the initial trajectory followed by the agent starting from a given point in the environment, and Figure 12 presents the activation level of some of its motives. This environment comes with *BugWorld*. The agent starts by reaching the upper left corner target and continues its path by following boundaries. It stops at the lower left corner where the charging station is located. It then continues to follow boundaries, reaching the lower right target and the upper right target successively. The FULFILLMENT motive is increased by 30% when a target is attained. After that, the agent reaches again the upper left target and the charging station. The EAT motive is fully activated when the agent is recharging, confirming that the behavior RECHARGE is used. At this point, the agent is able to detect similar sequences in its topological graph and to construct a loop in it. The agent has a higher level of CONFIDENCE and wants to EXPLORE. It starts exploring the center of the room just after leaving the charging station by using the TURN90 behavior at a point when the agent feels it is able to move away from the boundary without being influenced by other obstacles.

The agent continues to explore the environment this way, only using the TURN90 behavior at proper landmarks when it has never been used. After a while, the agent feels confident for long periods of time because it is able to locate itself in its topological graph and it cannot explore new sites in the environment. As illustrated in Figure 13, the motive

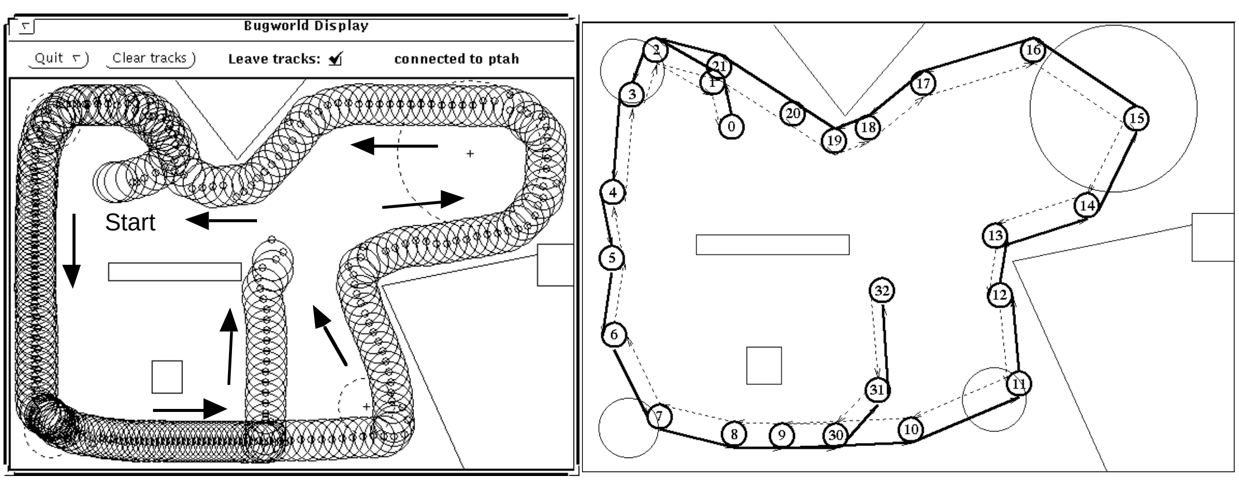


Figure 11: Trace and topological graph observed when the agent starts exploring the environment

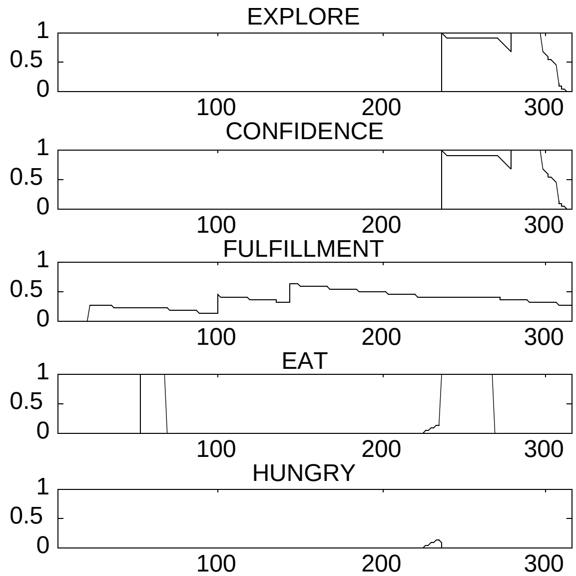


Figure 12: Important motives for the trace presented in Figure 11

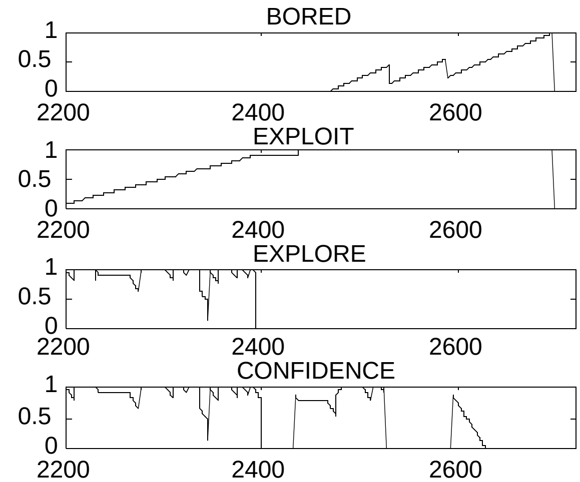
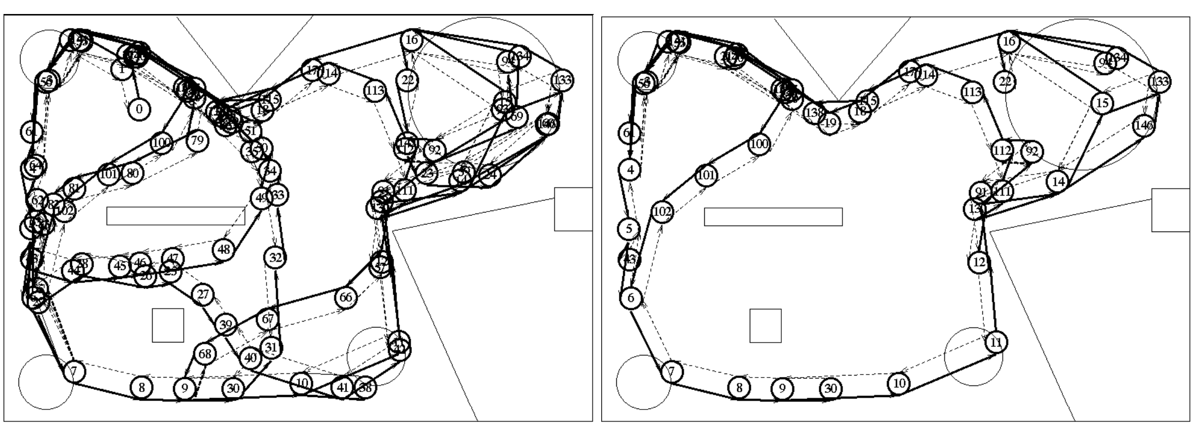


Figure 13: Motives when the topological graph is exploited

EXPLOIT increases until it reaches a preset value of 0.9. It then inhibits the EXPLORE motive, and the buffer nodes are used to locate the agent in its graph. The EXPLOIT motive reaches full activation when the agent arrives at a charging station and takes the time to optimize its topological graph. Then, the agent can use its graph to plan a path toward a target. In the case presented here, the agent is able to use its buffer nodes for positioning in the topological graph but at a certain point, the agent is unable to know where it is (as we can see by the zero CONFIDENCE level before and after the 2600 step). The agent is also unable to plan paths or to reach targets during that time, and the motive BORED increases until it resets the EXPLOIT motive.

Figure 14 shows the topological graph before and after optimization. As illustrated in a), because similarities between nodes are evaluated based on a sequence of similar nodes and not by factors uniquely identifying each possible landmark in the environment, the graph can be composed of parallel branches or multiple nodes for the same site. To resolve this possible confusion, we can see that it is important to optimize the graph. In b), the graph is reduced by approximately 50%. The paths following the boundaries of the environment are kept without being explicitly specified in the optimization procedure. This way, the useful paths emerge from the experiences of the agent in the environment and its abilities to use this representation.

In Figure 15, the agent starts from another point in the environment. Because of a conflict between EMERGENCY, AVOID and ALIGN behaviors, the agent gets stuck in the lower right corner. The simultaneous constant exploitation of EMERGENCY and AVOID excites the motive DISTRESS from which the BACKING behavior is recommended by the *Needs* module. The agent then starts moving towards the charging station, but observes a decrease in the exploitation of the TARGET behavior. This indicates that it is moving away from a target which, in this case, is the upper right target. The motive DECEPTION is then increased and the agent makes a U-turn by using the



a) Before optimization

b) After optimization

Figure 14: Topological graph before and after optimization

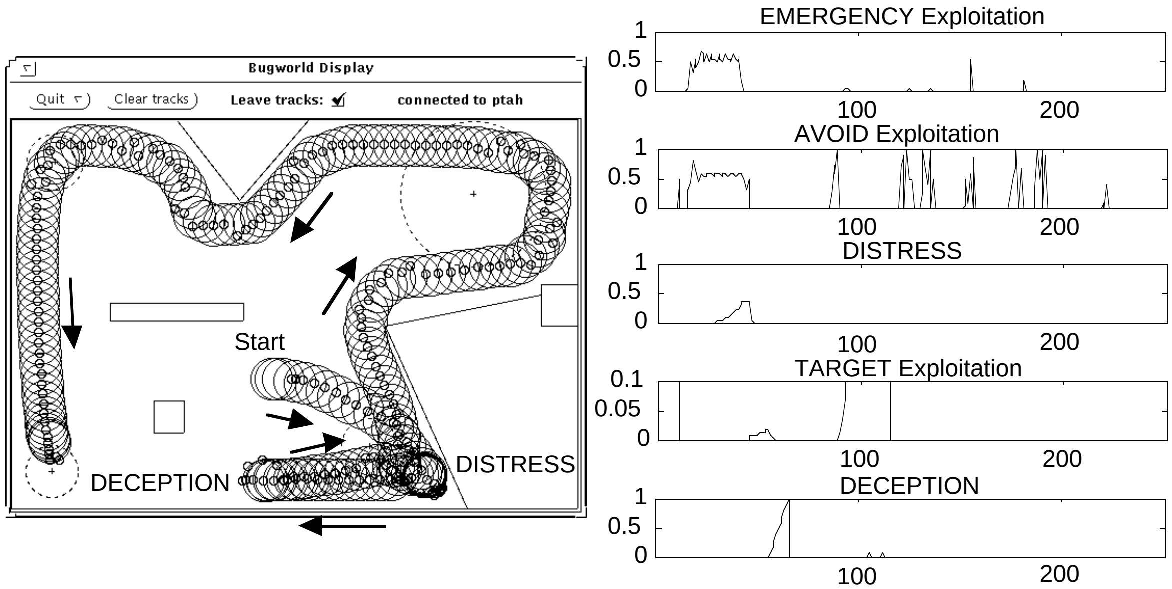


Figure 15: Trace and conditions for the motives DISTRESS and DECEPTION

TURN180 behavior. The agent continues its path by following boundaries until it reaches the charging station. An S.O.S. is emitted before arriving at this point because the agent has only three cycles of energy left.

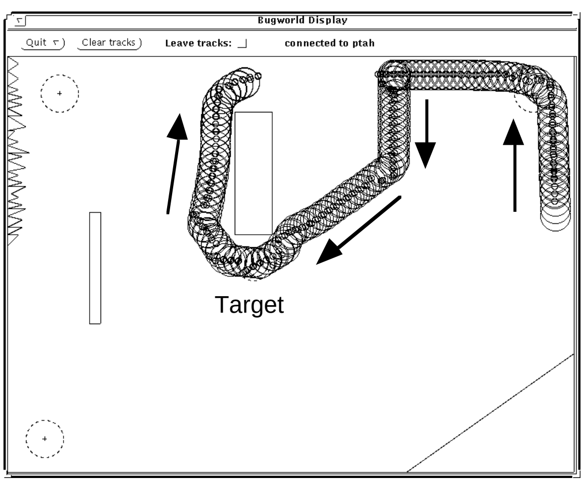


Figure 16: Path reproduction

Other environments have also been used during our experimentation. Figure 16 shows two parts of a trace made by the agent placed in another room. The first one is made when the agent explores the environment. The agent is then able to reach the target at the center of the room. Later, when the agent exploits its topological graph, it is able to plan a path toward this target and to reproduce the path using the buffer nodes and its optimized topological graph. Figure 17 shows a special condition which occurred when a moving obstacle was placed in the same room with the agent. At one point, the obstacle is moving toward the agent. The agent tries to move away from the obstacle, but cannot do so because its back side collided with the moving obstacle. The agent does not have any behaviors to avoid obstacles from its back. But, by observing that the SPEED behavior is fully exploited for a long period of time (because the agent wants to move but simply cannot get some speed), the motive DISTRESS is excited so that the BACKING behavior can be used.

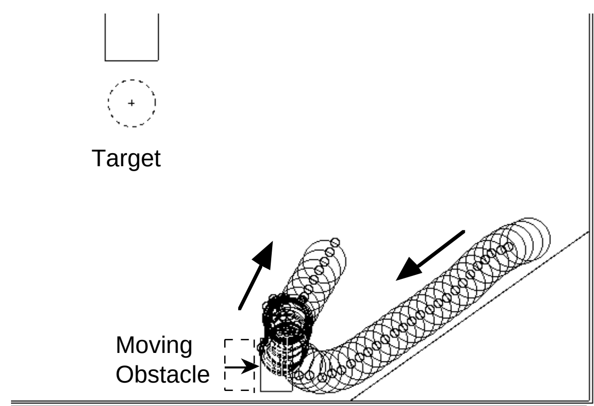


Figure 17: DISTRESS when a mobile obstacle came toward the agent

The emergence of functionality is very important in the mechanisms implemented for these experiments. Emergence is considered in behavior reactivity and parallelism, the fusion (or blending) of the control actions, behavior modification via the *Internal Parameters* link, and the dynamic selection of behaviors. Emergence is also considered for the fuzzy recommendation modules, the motives and the observation of *Behavior Exploitation*. Finally, the *Cognition* module and the topological graph considered emergence of the representation used by the agent.

This representation is constructed and managed directly from the actions and the internal capacities of the agent. Using this characteristic in all the architecture modules, the agent has showed its ability to adapt to the environment and to its own limitations of interacting with it.

5. Conclusion

This article is too brief to give all the details about the architecture proposed along with the description of the mechanisms used for the experimentation presented here, and references to related concepts from artificial or natural systems. The objective of the article is rather to introduce this architecture and to demonstrate the possibility and the usefulness of combining reactivity, planning, deliberation and motivation. This architecture is based on five hypothesis: intelligence is behavior-based; intelligence depends on the internal context and the external context; intelligence emerges holistically; introspection is a basic constituent of intelligence; and intelligence is affected by the autonomy of the system. All of these facts are important in making the agent adapt to its own reality. Using this architecture, an intelligent functionality emerges from the agent's interactions with its external environment and its internal intentional senses. Other types of mechanisms could be used in its modules and modules can be used, if needed, according to the purpose of the system to be controlled. This way, the architecture tries to unify the different views, principles, mechanisms and characteristics associated with intelligent behavior.

Acknowledgments

Support from the Natural Sciences and Engineering Research Council of Canada (NSERC) was highly appreciated. We would also like to thank Nikolaus Almàssy for letting us use *BugWorld*.

References

[Aho88] Aho, A.V., Sethi, R., and Ullman, J.D., *Compilers, Principles, Techniques, and Tools*, Addison Wesley, 1988.

[Albu91] Albus, J.S., “Outline for a theory of intelligence”, *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 473-509, May-June 1991.

[Alm93] Almàssy, N., “BugWorld: A distributed environment for the development of control architectures in multi-agent worlds”, Tech. Report 93.32 (ftp anonymous, almassy@ifi.unizh.ch), Dept. Computer Science, University Zurich-Irchel, December 1993.

[Beck83] Beck, R.C., *Motivation. Theories and Principles*, Second Edition, Prentice Hall, 1983.

[Beer90] Beer, R.D., *Intelligence as Adaptive Behavior. An experiment in computational neuroethology*, Academic Press, 1990.

[Broo86] Brooks, R.A., “A robust layered control system for a mobile robot”, *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 1, pp. 14-23, March 1986.

[Donn94] Donnart, J.Y. and Meyer, J.A., “A hierarchical classifier system implementing a motivationally autonomous animat”, in *Proc. Third Int'l Conf. on Simulation of Adaptive Behavior,* The MIT Press, 1994, pp. 144-153.

[Firb89] Firby, R.J., “Adaptive execution in complex dynamic worlds”, Ph.D. Thesis, Dept. Computer Science, Yale University, 1989.

[Haye95] Hayes-Roth, B., “An architecture for adaptive intelligent systems”, *Artificial Intelligence*, vol. 72, pp. 329-365, 1995.

[Kael86] Kaelbling, L.P., “An architecture for intelligent reactive systems”, in *Proc. Workshop on Reasoning about Actions and Plans*, 1986, pp. 395-410.

[Lee90] Lee, C.C., “Fuzzy logic in control systems: fuzzy logic controller”, *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 404-435, March-April 1990.

[Maes91] Maes, P., “A bottom-up mechanism for behavior selection in an artificial creature”, in *Proc. First Int'l Conf. on Simulation of Adaptive Behavior,* The MIT Press, 1991, pp. 238-246.

[Mata92] Mataric, M.J., “Integration of representation into goal-driven behavior-based robots”, *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 304-312, 1992.

[Mich96] Michaud, F., “Nouvelle architecture unifiée de contrôle intelligent par sélection intentionnelle de comportements”, Ph.D. Thesis, Dept. Electrical and Computer Engineering, Université de Sherbrooke, 1996 (in French).

[Nore95] Noreils, F.R. and Chatila, R.G., “Plan execution monitoring and control architecture for mobile robots”, *IEEE Trans. Robotics and Automation*, vol. 11, no. 2, pp. 255-266, April 1995.

[Saff93] Saffiotti, A., Ruspini, E., and Konolige, K., “A fuzzy controller for Flakey, an autonomous mobile robot”, Technical Note 529, SRI International, March 1993.

[Sari83] Saridis, G.N., “Intelligent robotic control”, *IEEE Trans. on Automatic Control*, vol. AC-28, no. 5, pp. 547-557, May 1983.

[Simm90] Simmons, R., Lin, L.J., and Fedor, C., “Autonomous task control for mobile robots”, in *Proc. Fifth Int'l Symposium on Intelligent Control,* 1990, pp. 663-668.

[Simo95] Simon, H.A., “Explaining the ineffable: AI on the topics of intuition, insight and inspiration”, in *Proc. Int'l Conf. Artificial Intelligence (IJCAI)*, Montréal, August 1995, pp. 939-948.

[Wern92] Werner, E., “The design of multi-agent systems”, in *Proc. Third European Workshop on Modeling Autonomous Agents in a Multi-Agent World*, Elsevier Science, 1992, pp. 3-28.