IEEE Int'l Conf. on Fuzzy Systems, Barcelona, Spain, July 1997

Selecting Behaviors using Fuzzy Logic

François Michaud

Interaction Lab
Computer Science Department
Volen Center for Complex Systems
Brandeis University
Waltham MA 02254
michaudf@cs.brandeis.edu
http://www.cs.brandeis.edu/˜michaudf/

Department of Electrical and Computer Engineering
Université de Sherbrooke
Sherbrooke (Québec CANADA) J1K 2R1
michaudf@gel.usherb.ca

Abstract

Behavior-based systems, i.e. systems that use behaviors as a way of decomposing the control policy needed for accomplishing a task, are very useful in making robots cope with the dynamics of real-world environments. However, these systems still need to be extended to design robots having multiple and possibly conflicting goals, requiring planning, and getting more out of their behaviors. One way of doing that is to allow behaviors to be selected dynamically and to blend their actions in order to get more complex behaviors. This paper addresses these issues by presenting a control architecture that, when using fuzzy logic, allows behaviors to be efficiently selected by different sources. A simulated world for mobile robots is used to illustrate these ideas.

Key words: Behavior-based systems, fuzzy selection of behaviors, fuzzy behaviors, control architecture.

1. Introduction

Behavior-based systems, as introduced by Brooks [4], have proven to be very useful in making robots cope with the dynamics of real-world environments. A behavior-based system uses behaviors as a way of decomposing the control policy needed for accomplishing a task. The behavior repertoire defines the system's skills for reacting to the situations encountered in its environment. In Brooks' subsumption architecture, behaviors all run in parallel and a fixed and predefined hierarchy is used where only one behavior at all times has predominance over the others.

However, extensions to this architecture are needed to allow the system to handle more complex tasks requiring greater planning abilities or for managing multiple and possibly conflicting goals. To do that, one interesting methodology is to dynamically select behaviors by activating them according to the intentions of the system and by modifying their relative importance [10,11,12]. Also, to get more out of the activated behaviors, their actions may be blended to derive different control actions that take into consideration their respective goal and their current priority.

Fuzzy logic has been successfully used with a robot by Saffiotti *et al.* [13,14,15] to select behaviors and to combine their control actions. The blending of the control actions of the behaviors allows smooth transitions between behaviors and can lead to more complex emerging behaviors. In this system, behaviors are activated according to a desirability measure obtained from a goal-directed planner. However, this planner is responsive to commands given by a user, and to have a more autonomous system other modules might be required to activate behaviors.

As an extension to this mechanism, this paper proposes a recommendation mechanism using desirability and undesirability measures. Three modules are used to recommend behaviors, and these measures are very effective in handling possible conflicts in selecting behaviors from these sources. This paper also shows that the observation of the exploitation of behaviors (i.e. their actual use to control the system) can give important information concerning the efficiency of the system in behaving in its environment. These mechanisms have been validated using a simulated environment for mobile robots, in the context of studying the characteristics of a new control architecture for designing intelligent and autonomous agents [10,11,12].

2. Control architecture for dynamic selection of behaviors

The primary objective of this research was to design a unified control architecture capable of combining the interesting properties associated to "intelligence", like reactivity, planning, deliberation and motivation, while preserving their underlying principles [10,11]. This architecture, illustrated by Figure 1, is built on top of behaviors, and the idea is to select these behaviors and to change their respective priority to make the system behave appropriately according to the situations it encounters in the world. The relevance in using each behavior is determined by the *Recommendation Level*. Finally, the *Motivation Level* is responsible for coordinating and supervising the agent's goals according to its actual experiences in the environment.

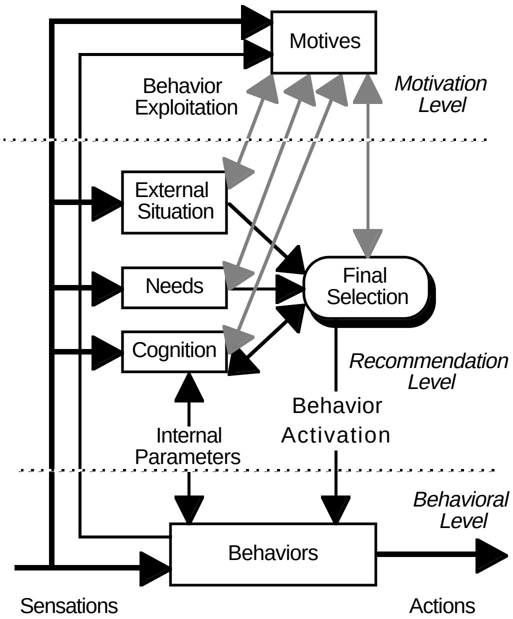

Figure 1. Control architecture

Different AI methodologies can be used to implement the modules of this new control architecture. Note also that the usefulness of the modules or of the links between them depends on the requirements of the task. In the following sections, the description of the fuzzy implementation for recommending and selecting behaviors is given, followed by the results obtained using *BugWorld* [1], a simulated environment for mobile robots. In the rest of the paper, the term "agent" will refer to a system (like a real or a simulated robot) that is using this architecture to control its actions.

3. Characteristics of the simulated mobile robot and the task

To validate the properties of the control architecture and its fuzzy implementation, it was required to develop an autonomous agent that has to deal with various goals like managing its energy, its purposes and its well being. To do this, a simulated world for mobile robots called *BugWorld* [1] was used. In the 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 has a limited memory to acquire knowledge that can be helpful in its task. For sensing, the agent has at its disposal eight directed proximity sensors for obstacle, each separated by 45° starting from its nose. There are also two undirected target sensors, one on each side. Note that the proximity sensors and the target sensors do not model any existing physical sensing devices. 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. For actions, the agent can control the speed, the rotation and a variable for the color of the agent.

To achieve this task, the agent is designed 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.

4. Fuzzy recommendation level

For this task, twelve behaviors were designed to control the agent: EMERGENCY for moving the agent when immediate danger is detected in its front; AVOID to move away from obstacles; SPEED to maintain a constant cruising velocity; ALIGN to follow boundaries; TARGET to align the robot toward a sensed 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 [10,11]).

The relevance in using each behavior is determined by the context, and this context is affected by what can be sensed in the environment and the internal states of the agent. Psychology indicates that human behavior is affected by the environment, the needs of the individual and the knowledge acquired or innate about the world [5]. Based on this decomposition, the control architecture uses three recommendation modules to affect the selection of behaviors. 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 based on knowledge learned or innate about the environment or the effect of the agent in its world. The influences of each of these modules may serve can be assimilated to a global behavior more reactive, egoistic, or rational.

These three modules recommend the use or the inhibition of behaviors by using a desirability measure and an undesirability measure for each behavior. 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 [2]. This differs from the approach of Saffiotti *et al.* [13,14,15] which uses the notion of desirability to describe the control function of a behavior. In the architecture proposed, these measures make it possible to prevent possible conflicts when recommending behaviors. These conflicts may occur inside the same recommendation module or from the parallel evaluation of these three recommendation modules.

```
<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 2. 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 3. Rules for the *Needs* module

Fuzzy logic can be used in these modules to recommend behaviors. For example, Figure 2 shows the fuzzy rules used by the *External Situation* module. In these rules, an undesired behavior is a consequence preceded by NOT. The *Needs* module uses a similar process to select behaviors but instead of only using external conditions, it considers also fuzzy states derived from the motives (see Section 6) to recommend behaviors. Figure 3 presents these rules.

The *Cognition* module differs from the two other recommendation modules in that it does not use fuzzy logic to formulate its recommendation. In the experiment done with the simulated world for mobile robots, the *Cognition* module is used to build a topological map of the environment. This topological map represents the world in a graph using topological information deduced from the proximity of obstacle at the front, at the back and at the sides of the agent. Examples of typical nodes used in this graph are *Right side*, *Corridor*, *Internal corner of 90˚ to turn*, etc. This topological representation is built using only the information coming from the presence or absence of an obstacle in the proximity of the agent, and the sequence of this identification process in time; it is not influenced by any global positioning coordinates. The purpose of this paper is not to describe the characteristics of this representation mechanism, but to illustrate that other types of knowledge can be integrated in the control architecture proposed for recommending behaviors. A more complete description of this algorithm can be found in [10,11].

The desirability and undesirability measures coming from these recommendation modules are then processed by the *Final Selection* module, which combines them appropriately to establish the activation of the behaviors (*Behavior Activation*). In the fuzzy implementation of the control architecture, *Behavior Activation* is evaluated based on a hedonic continuum [2] established from the fuzzy desirability μ_{des} and undesirability μ_{und} measures. First, these measures are respectively combined for each behavior using the fuzzy disjunction operator $\oplus$ maximum. Then, the combined undesirability is subtracted from the combined desirability measure and the behavior is activated if the result is greater than zero. Relation (1) shows these operations for behavior j and where m represents the recommendations from the three recommendation modules. So, to be activated, the desirability of a behavior must be higher than its undesirability.

$$\mu_{act}(j) = \max\left(0, \oplus\left[\mu_{des_m}(j)\right] - \oplus\left[\mu_{und_m}(j)\right]\right) \quad (1)$$

5. Fuzzy behavioral level

An increasing number of works now use fuzzy logic for implementing behaviors, like [3,6,8,13-16]. Two major reasons for using fuzzy behaviors are that they are very useful for quantifying analog values and establishing a smooth and efficient control from these values, and because they allow the blending of control actions formulated by the parallel use of behaviors. This is a nice extension to the subsumption architecture [4] where only one behavior is used to control the agent at all times, following a fix priority hierarchy between the behaviors. Fuzzy logic in this case allows the simultaneous contribution of behaviors, adapting them more efficiently to the particular but combined purpose of each behavior activated to control the agent.

A fuzzy behavior uses rules and linguistic variables to establish the correspondence between sensations and actions. The processing steps are similar to the ones for fuzzy systems [7], which are fuzzification, rule inference and defuzzification. The fuzzy conjunction operator $\otimes$ used is the minimum, and the fuzzy disjunction operator $\oplus$ used is the maximum. Defuzzification is done by using the center of area method [7], allowing the blending of the actions given by the behaviors. The only difference with the fuzzy inference steps is that rule firing strength μ_{Bx} is affected by the μ_{act}, the activation of the behaviors, given by the *Final Selection* module (see relation (1)). This operation is presented by relation (2) for rule r of the behavior j. Linguistic variables B and C refer to fuzzy variables associated with a control variable.

$$\mu_{C_{rj}}(Action) = \otimes\left[\mu_{B_{rj}}(Action), \mu_{act}(j)\right] \qquad (2)$$

<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 4. Rules for the EMERGENCY behavior

Figure 4 presents the rules of one behavior called EMERGENCY, and Figure 5 shows the membership functions for this behavior. 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 front of it.

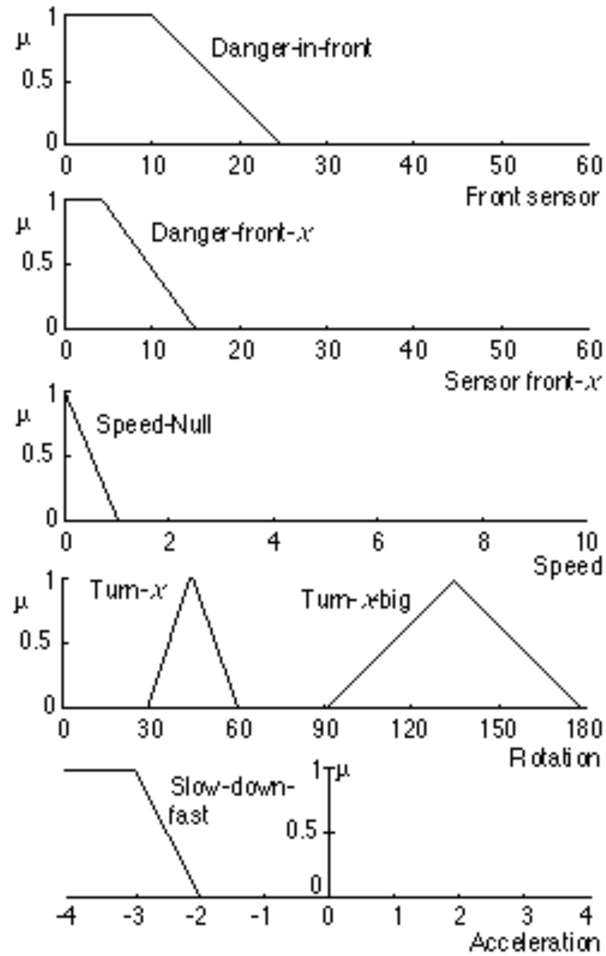

Figure 5. Membership functions used by the EMERGENCY behavior

6. Fuzzy exploitation of behaviors and *Motives*

Another interesting source of information available to the agent to characterize its behavior in its environment comes from self-observation of the exploitation of its behaviors. 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). Because behaviors are fuzzy in this implementation, *Behavior Exploitation* is a fuzzy measure defined in relation (3), approximating the contribution or the importance of the behavior to the fuzzy control actions formulated before defuzzification. It combines the activation of a behavior with its reactivity to the environment.

$$\mu_{exp}(j) = \mu_{act}(j) \otimes \left(\oplus\left[\mu_{B_{rj}}(Action)\right]\right) \qquad (3)$$

One particular influence coming from these observations is for affecting the motives of the agent. In the proposed control architecture, the *Motives* module is used to affect the recommendation of behaviors, to coordinate actions between modules and to influence the importance of the recommendation modules if necessary. Observing the exploitation of behaviors can help the agent verify that its intentions (reflected by its internal state and the behaviors activated) are getting correctly satisfied.

The agent has five basic goals in the environment, and ten motives were designed to handle these goals [10,11]. Activation variables as in Maes [9] are used for motives. The motives HUNGRY and EAT decide when the agent must find a charging station and recharge itself; the motive FULFILLMENT determines when to reach targets; and the motives CONFIDENCE, CERTAINTY, EXPLORE, EXPLOIT and BORED manage the influences of the topological graph built by the *Cognition* module. Finally, two motives are used to monitor the good operation of the agent, and these motives are particularly influenced by the *Behavior Exploitation* measure. 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 SPEED behavior is composed of only two rules indicating when to increase or to decrease the speed of the agent. These rules use two fuzzy linguistic variables overlapping by 50% at the desired velocity, so a constant behavior exploitation of 0.5 indicates normal working conditions. DISTRESS influences the use of the BACKUP behavior via the *Needs* module. The other motive, called DECEPTION, increases when the agent is moving away from a sensed target or charging station. This is 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.

7. Experimental results

To illustrate the concepts described previously, this section presents trajectories followed by the agent in various situations, along with the decision variables explaining its manifested behavior.

Figure 6 illustrates the initial trajectory followed by the agent starting from a given point in the environment. 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. After that, the agent again reaches the upper left target and the charging station. At this point, the agent is able to position itself in its topological graph (shown on the right, and built as the agent explores its environment), and decides that it is time to start exploring other areas of the room. 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.

Recommendations for behavior TARGET during this trace are presented in Figure 7. The *External Situation* module inhibits the use of TARGET when an obstacle is detected in front of the agent (see rule *Obstacle* of Figure 2). The *Needs* module is in favor of using TARGET when the motive FULFILLMENT is small, but not when

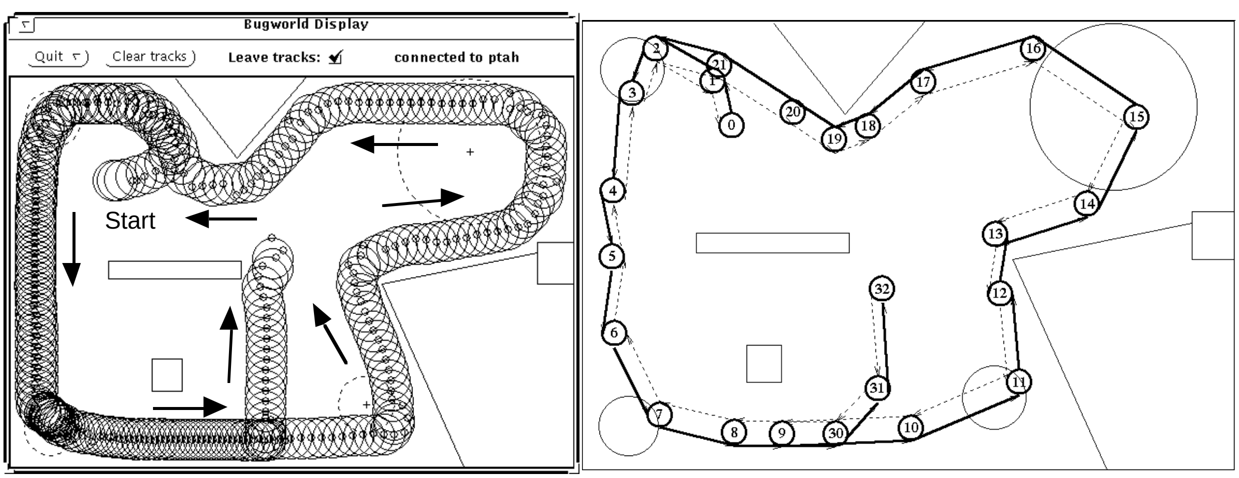

Figure 6. Trace and topological graph observed when the agent starts exploring the environment

the agent is recharging. The global recommendations are obtained from the fuzzy disjunction of respectively the desirability and undesirability measures formulated by the recommendation modules. The activation of TARGET is the result of relation (1).

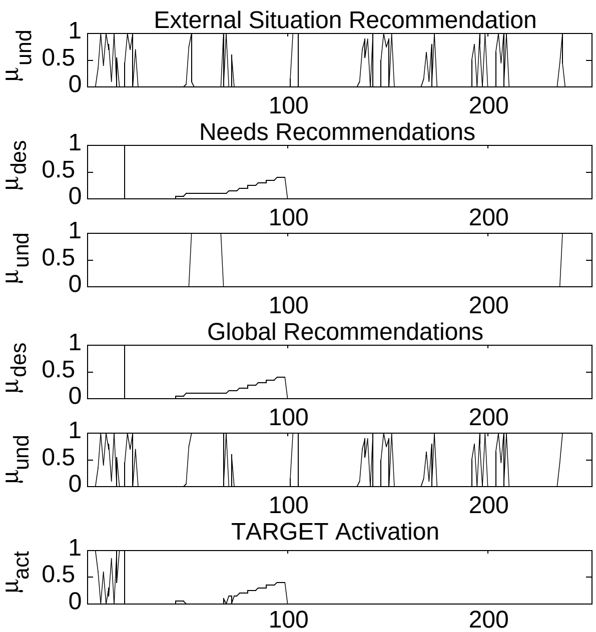

Figure 7. Recommendations for behavior TARGET

In Figure 8, the agent starts from another point in the environment, leading directly into the lower right corner. Because of a conflict between EMERGENCY, AVOID and ALIGN behaviors, the agent gets stuck in this corner. The simultaneous constant exploitation of EMERGENCY and AVOID excites the motive DISTRESS from which the BACKING behavior is recommended by the *Needs* module. This example clearly indicates that observing the exploitation of behaviors in time may be very useful in managing conflicts between behaviors at a larger time scale than only using the desirability and undesirability measures. The desirability/undesirability measures are more useful to handle conflicts for the current activation of behaviors. Motives influenced this way can be used as a memoriza-tion mechanism for past events that can affect the present selection of behaviors. It is very important in making a control architecture (specially if it is based on behaviors) more than only reactive, and in making the agent more autonomous.

After having backed up away of the corner, the agent 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 TURN180 behavior. The agent continues its path by following boundaries until it reaches the charging station. This shows how the observation of behavior exploitation can serve as a useful indication of the progress of the agent in satisfying its intention of reaching targets.

A similar situation of dissatisfied intention happened when the agent was placed in another room having a moving obstacle in it. Figure 9 shows the obstacle moving toward the agent. The agent tries to move away from it but could not do so because its back side collides

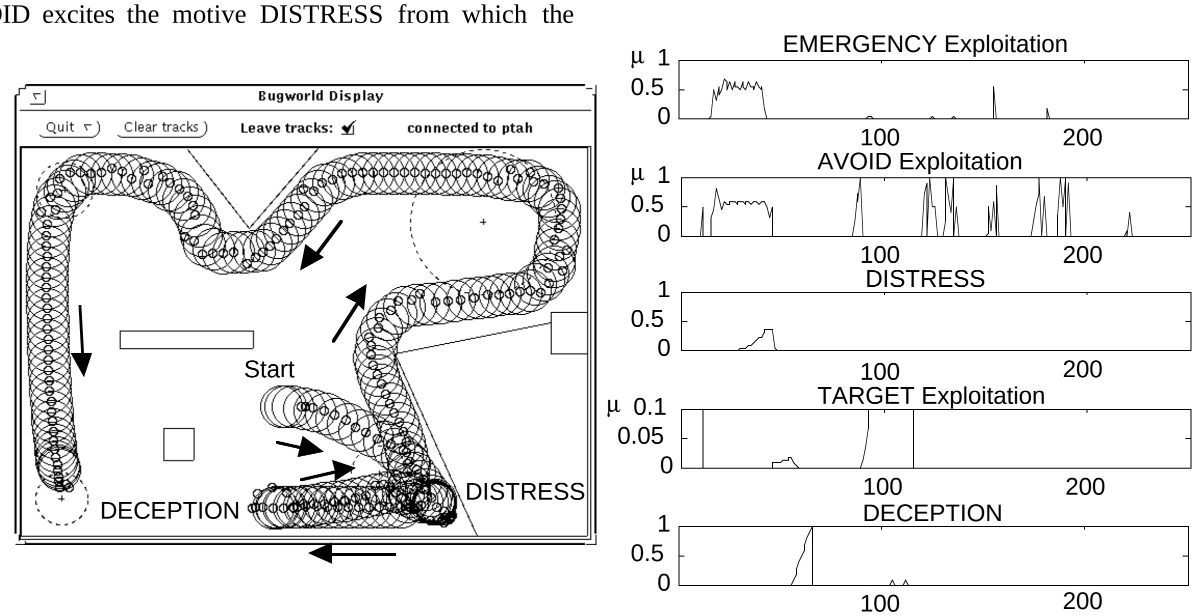

Figure 8. Trace and conditions for the motives DISTRESS and DECEPTION

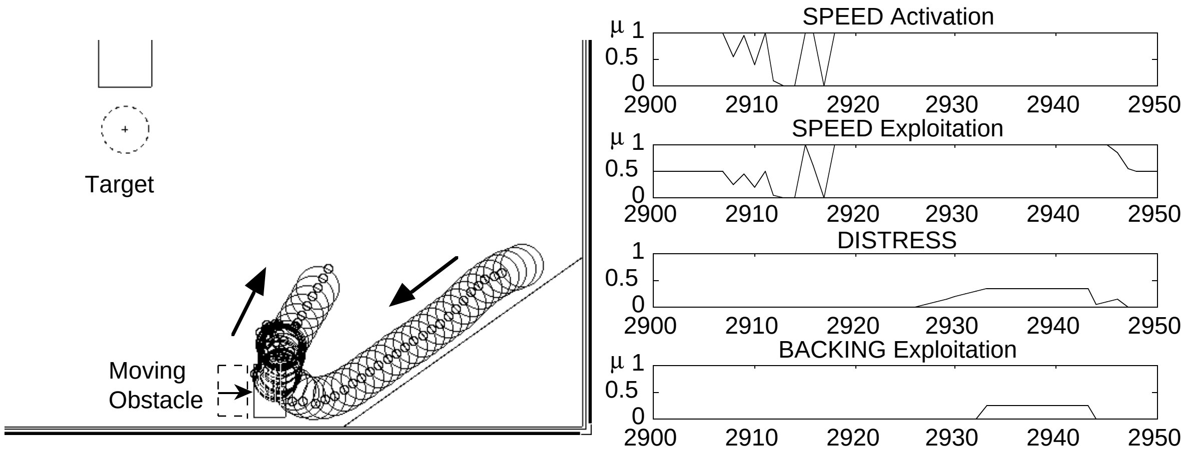

Figure 9. DISTRESS motive when a mobile obstacle came toward the agent

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 ahead but simply cannot get some speed), the motive DISTRESS is excited so that the BACKING behavior can be used.

8. Discussion

These results show the ability of the agent to efficiently interact with its environment by being able to adapt to the encountered situations, to its goals and to its acquired knowledge. However, at lot of expertise must be transmitted by the designer to the agent by programming the different modules. The choice of behaviors to make the behavior repertoire of the agent, the recommendation conditions and the choice of motives have a profound effect on the way the agent will behave in the environment. This is also particularly true using fuzzy logic because the designer has to design the rules and to adjust the membership functions for the behaviors and the other fuzzy modules. Even though these adjustments can be learned, they are not that hard to do if a good design methodology is used.

For these experiments, the design of the behaviors and the other mechanisms used in the proposed control architecture did not follow a top-down or a bottom-up approach; it relied instead on an increasing 'level of intelligence' principle. First, EMERGENCY, AVOID, SPEED and ALIGN behaviors, activated by the *External Situation* module, were designed to make the agent move in the environment by following boundaries. Then, the behavior TARGET with the motive FULFILLMENT, along with the activation rule in the *Needs* module, were added to make the agent find targets. Other design strategies in case of distress, for recharging, exploring, exploiting the topological graph, and making a U-turn were also implemented and tested using this methodology.

Following these design strategies according to an increasing 'level of intelligence' simplified the overall interactions between the modules of the architecture. First, it helped identify possible conflicts between the intentions of the agent. These conflicts can be managed during the design of behaviors, the desirability and undesirability measures, and the influences on motives. As shown in the previous section, observing the exploitation of behaviors makes the agent aware of how it behaves in the world. By comparing these observations with the goal of the behaviors, it also helps manage unexpected behavior situations not anticipated by the designer. This plays an important part in making the agent adapt to its own abilities of interacting with its environment.

Secondly, following the design strategies also helped to adjust the membership functions used by the behaviors to get smooth and consistent control actions. Rule design and the choice of the fuzzy consequences used by the behaviors has also an effect on the simultaneous use of behaviors. One influence is that to have a compact control mechanism for the agent, it may be a good thing to use the same fuzzy consequences between behaviors. But because of the unification of all the fuzzy consequences before defuzzification, it is possible to loose the control action formulated by one behavior because another one is using the same consequence. For example, if a behavior gives a command of *Turn-left* with membership value of 0.5, and another one gives a command of *Turn-left* with membership value of 0.8 and also *Turn-right* with membership value of 1, then the action of the first behavior will not be taken into consideration. If the reactive conditions of the first behavior represent very particular and important conditions that must be considered when they are formulated, then the use of another fuzzy consequence may be required. When it is possible, adjusting the respective behavior activation level for giving priority to the first behavior can also help to handle these problems [11].

9 . Conclusion

The main focus of this paper is to show how fuzzy logic is used in a new control architecture to extend the capabilities related to behavior-based systems. The first extension concerns the ability to recommend behaviors by using multiple and independent modules. One direct analogy for characterizing the recommendation level of the proposed control architecture is to say that it is composed of "behaviors" (*External Situation*, *Needs* and *Cognition*) and an arbitration mechanism (*Final Selection*) for recommending behaviors. The use of desirability and undesirability measures, formulated according to a hedonic continuum, makes it possible to manage conflicts between these recommendation modules. These modules can also use different mechanisms for making these choices without causing problems to the selection process, as it is demonstrated in the experiments by using a topological graph in the *Cognition* module.

The second extension is to use a fuzzy exploitation measure to make the agent observe its use of the behaviors activated for controlling its actions. Observing the exploitation of behaviors makes the agent aware of how it behaves in the world. By comparing these observations with the goal and the proper working of the behaviors, it also helps manage unexpected behavior situations not anticipated by the designer. This plays an important part in making the agent adapt to its own abilities of interacting with its environment. Along with the first extension, this also results in having the selection of behavior emerge from the experiences of the agent in its environment and their impact on its intentions.

The third extension is that fuzzy behaviors are better suited to handle analog sensations and provide an efficient mechanism for blending control actions coming from multiple behaviors exploited at the same time. Overall, these extensions have the effect of making agents more efficient and more autonomous.

Finally, this control architecture was also validated using a *Rug Warrior* robot (see http://www.tiac.net/users/akpeters/Rug-warrior.html), but without using fuzzy logic. We still hope to implement the mechanisms described in this paper using a real robot in the near future.

Acknowledgments

This research was done at the Department of Electrical and Computer Engineering of the Université de Sherbrooke, Québec, Canada. Support from the Natural Sciences and Engineering Research Council of Canada (NSERC) was highly appreciated. The author would also like to acknowledge the participation of Gérard Lachiver and Chon Tam Le Dinh to this project, and thank Nikolaus Almàssy for making *BugWorld* available for experimentation.

References

[1] 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), Department of Computer Science, University Zurich-Irchel, December 1993.

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

[3] Bonarini, A., "Reinforcement learning of hierarchical fuzzy behaviors for autonomous agents", *Proc. IPMU*, 1996, pp. 1223-1228.

[4] 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.

[5] Dolan, S.L. and Lamoureux, G., *Initiation à la psychologie du travail*. Gaëtan Morin Ed., 1990.

[6] Goodridge, S.G., Luo, R.C., and Kay, M.G., "Multi-layered fuzzy behavior fusion for real-time control of systems with many sensors", in *Proc. IEEE Int'l Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Las Vegas, October 1994, pp. 272-279.

[7] 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.

[8] Li, W., "Fuzzy logic-based 'perception-action' behavior control of a mobile robot in uncertain environments", in *Proc. IEEE Int'l Conf. on Fuzzy Systems*, 1994, pp. 1626-1631.

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

[10] Michaud, F., "Nouvelle architecture unifiée de contrôle intelligent par sélection intentionnelle de comportements", Ph.D. Thesis, Department of Electrical and Computer Engineering, Université de Sherbrooke, May 1996.

[11] Michaud, F., Lachiver, G., and Le Dinh, C.T., "A new control architecture combining reactivity, deliberation and motivation for situated autonomous agent", in *Proc. Conf. Simulation of Adaptive Behavior*, P. Maes, M. J. Mataric, J.-A. Meyer, J. Pollack, S. W. Wilson (eds), September 1996.

[12] Michaud, F., Lachiver, G., and Le Dinh, C.T., "Fuzzy selection and blending of behaviors for situated autonomous agent", in *Proc. IEEE Int'l Conf. on Fuzzy Systems*, September 1996.

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

[14] Saffiotti, A., Konolige, K. and Ruspini, E., "Blending reactivity and goal-directedness in a fuzzy controller", in *Proc. IEEE Int'l Conf. on Fuzzy Systems*, 1993, pp. 134-139.

[15] Saffiotti, A., Konolige, K. and Ruspini, E., "A multivalued-logic approach to integrating planning and control", *Artificial Intelligence*, vol. 76, no. 1-2, 1995, pp. 481-526.

[16] Surmann, H., Huser, J. and Peters, L., "A fuzzy system for indoor mobile robot navigation", *Proc. IEEE Int'l Conf. on Fuzzy Systems*, 1995, pp. 83-88.