Action Selection in an Autonomous Agent with a Hierarchical Distributed Reactive Planning Architecture

Vincent DECUGIS† & Jacques FERBER ‡

† Groupe d'Etudes Sous-Marine de l'Atlantique (GESMA),
BP 42 , 29240 BREST NAVAL

‡ Laboratoire d'Informatique, Robotique et Microélectronique de Montpellier
161, rue Ada, 34392 MONTPELLIER Cedex 2

Abstract

Action selection is a central issue for robotic autonomous agent design. The action selection mechanism used in an autonomous robot needs to be reactive, but must include planning capacities. It should enable the coexistence of antinomic goals and constraints, and induce a sufficient adaptivity to cope with circumstances unexpected at design time. To our knowledge, no such mechanism embedding all these properties has been implemented in a robotic application. We propose here an action selection mechanism that fulfill all these requirements and present a principle test of its properties on a simple robot realistic simulator.

1 Introduction

A wide panel of solutions have already been proposed to solve the problem of action selection in autonomous agents, especially for software agents in which real-time constraints are relatively loose. Yet, in physical autonomous agents such as autonomous mobile robots, action selection is still a difficult problem which usually influences the whole software architecture of the robot and determines its efficiency.

Robotics applications induce some requirements for action selection in autonomous agents :

- *reactivity*; the agent needs to choose its action or behavior as quickly as possible when some change occurs in the environment. Reactivity permits the short term best choice.
- *planning*; the agent should be able to predict the consequences of its actions, and take these into account for choosing its behavior. When time is available, it should prepare a series of actions to be undertaken in the future. Planning enables to orient choices towards preset goals, enabling the agent to fulfill a useful mission. Planning permits the long term best choice. It is usually a time-consuming process, thus rather incompatible with reactivity.
- *incompatible goal management*; robotics agents face commonly situations where incompatible goals and constraints conflict : reaching some spot and avoiding an obstacle, refuelling its battery and doing its job, etc. Action selection mechanism must manage properly these incompatibilities.
- *adaptivity*; it is often impossible for the designer of a robotic agent to predict all the situations it will face. Adaptivity should modify the process of action selection to adapt these unexpected circumstances and modify the agent's knowledge about its actions.

Maes proposed an ASM in [13] and [14] that enables the combination of reactivity and planning. For achieving this difficult mixing, Maes relaxes the constraint to build a plan, and take into account at the same level the present situation of the agent and the expected future situations depending on its choices. This is for the action selection problem a better solution than time constrained planning approaches, such as reactive or anytime planning ([6],[7],[8],[11], [10]), that focus on the building of a plan. Extension of Maes' work in [15] introduced adaptivity properties in the ASM, enabling the agent to learn the consequences of its actions. We found two limitations in the use of this ASM. The first is linked to the conflicting goal management requirement. Tyrrell showed in [23] that Maes' ASM was not able to fulfill antinomic constraints in a simple experiment. An agent implemented with Maes' ASM was instructed to both go at one place for "eating" and to another place to "drink". It constantly changed its predominant goals, wandering between the two spots while never reaching any of them. Secondly, perception of the environment in [13] is done through the use of symbols, and actions are treated as symbolic rules encapsulating an open-loop instantaneous influence on the environment.

We thought that the first problem can be solved by the use of hierarchical ASMs found in ethology. Tyrrell gives in [22] a review of these mechanisms proposed by Lorenz [12], Tinbergen [21] and Baerends [1]. They are all based on a hierarchical organization of the drives of an animal. Each node of the hierarchy is viewed as controlling a set of typical behaviors. Its function is to arbitrate, when it is active, the choices between these possible behaviors. At the bottom of the hierarchy, basic reflexes are found, such as reflex movements, orientation and basic perceptive mechanisms; at the top, the great drives of animals such as feeding, reproducing and surviving; and in the intermediate levels, complex behaviors such as chasing, preening, building a nest, etc. Such

organizations are perfectly suited for managing conflicting goals. What lacks in those ASMs is a descriptions of the arbitration mechanism at nodes of the hierarchy. This is precisely where we can use a "flat" ASM[1] such as Maes' mechanism.

We bypass the second problem by adapting perceptions and actions used in Maes's mechanism to a real robotic situation. We therefore use in the flat ASM only situated symbolic information which can be easily computed from raw sensor data. Following a situated approach [18], the actions are no longer instantaneous and can be reflex behaviors, direct sensory-motor coupling or simple servo loops or control laws. This leaded us to some modifications of the original Maes' ASM.

2 Description of the flat action selection mechanism

We describe now this adaptation of Maes' ASM, which constitutes for us the "flat ASM", building block of our global hierarchical framework ASM described in the next section.

2.1 Components of the mechanism

The flat ASM is composed of a set of two different kind of components organized in an interacting network:

- *perception components.* Perceptions are predicates over the sensor space $\mathcal{S}$, i.e. the set of all possible sensor values. If the agent has two sensors whose values are taken in sets $\mathcal{S}_1$ and $\mathcal{S}_2$, then the agent sensor space will be $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2$. So perception component P truth value is a function $v_P : \mathcal{S} \longrightarrow \{0, 1\}$. The set of all perceptions is noted $\mathcal{P}$. We pose $card(\mathcal{P}) = n_P$, and note the different perceptions P_i with $1 \leq i \leq n_P$.

- *behavior components.* If $\mathcal{E}$ is the set of all possible motor actions of the robot – the effector space –, then a behavior in its whole generality can be defined as a function from $\mathcal{S}$ to $\mathcal{E}$, i.e. a system that links sensory inputs to motor outputs. The set of all behavior components is noted $\mathcal{B}$. We pose $card(\mathcal{B}) = n_B$, and note the different behaviors B_i with $1 \leq i \leq n_B$.

Each of these components are given a scalar characteristic called *activation level*. The interpretation of activation is slightly different between perceptions and behaviors :

- for behavior component B , activation level α_B represents the utility of the behavior in the actual situation of the robot to achieve its goal. The more α_B is high, the more component B is likely to be chosen by the network.

- for perception component P, activation level α_P represents both the propensity of this perception to be or become true in the near future, or the "desire" of the network to make it become true.

The interactions between the components of the architecture are made through links. The function of these links is to transmit a flow of activation from one component to another. The links are directional : a source and a target component are identified for each of them, and their roles are different. Nonetheless, activation flows through the link in both directions. This means that the flow is asymmetric in the two directions. Each link have a "strength" which determines its resistance to activation flow. A link with a strength of 1 will let activation flows freely, whereas a link with strength 0 would not let it flow at all, and is equivalent to have no link. We note f_{PB} the strength of a link from perception P to behavior B and f_{BP} the strength of a link from behavior B to perception P.

Links have a precise interpretation in the network :

- when there is a *link from a perception to a behavior*, this perception truth value is significant for the choice of that behavior. If the perception is actually true, then the target behavior should have a higher probability to be chosen by the network. The stronger the link, the more perception will influence the choice of the target behavior;

- when there is a *link from a behavior to a perception*, this perception is likely to become true after the use of the source behavior. The stronger the link, the more probable the perception will become true.

Perception to behavior links can be mandatory. When a behavior is the target of one of this kind of link, he cannot be selected by the network as long as the source perception has not a true truth value.

The topology of the network, i.e. the list of links, is determined by the designer of the ASM to suit his goals. The way to achieve a particular behavior by fixing links strength and topology will be explained in the experiment section.

2.2 Evolution mechanism of activation levels

In usual conditions, the network will receive a constant flow of activation from two sources :

- *perception components whose truth value is true.* This reflects the influence of the actual situation of the agent in its environment. By using perception truth values, activation levels take this situation permanently into account. The amount of activation introduced is constant and arbitrarily fixed to 1.

- *goals.* Goals are constant activation sources pointing to one particular component, most often perception. Pointing a goal to a perception will tend to influence the actions of the agent so that it becomes true. Pointing to a behavior will indicate that we want this behavior to be chosen, and the network will tend to make its mandatory conditions becoming true. The amount of activation introduced by goal G will be noted α_G.

There is no a priori restriction of putting several goals in the network, though we will see this is not an efficient design policy. We will take nonetheless this possibility into account in our formal modelling of the mechanism, since it imposes no stronger restriction.

We had specified that activation is flowing in both ways of a link. We will say it flows in the direct way if its direction matches the link direction, and that it flows in the indirect way in the opposite case. Direct and indirect activation propagations have two clear different interpretations :

- *direct propagation* represents the influence of the current situation of the environment on the network. True perceptions receive activation, which propagates to some behaviors that become thus more likely to be selected. These behaviors propagate their activation to

[1] according to Tyrrell's terminology, flat means that the actions are considered all at the same level, by opposition to a hierarchical organization.

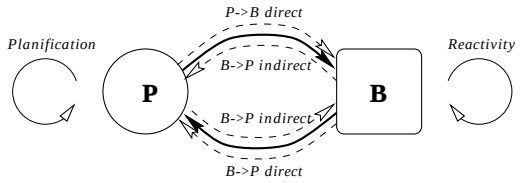

Figure 1: *Propagation of activation in both direction in links of the network.*

other perceptions through behavior to perception links whom they are sources. These target perceptions are therefore more likely to become true, and propagate themselves activation to their target behaviors, and so on. So, from one true perception, activation will diffuse through the network, reflecting the consequences of this true value and permitting it to take into account to variation of the environment state.

- *indirect propagation* enables the network to influence action choice towards the goals. One goal will introduce activation to a component, perception for instance. This perception will use links whom it is the target to retro-propagate its activation level. This will increase activation level in behaviors whose probable consequences will be to make the goal perception become true. These behaviors will themselves retro-propagate their activation to perceptions which will help them to be selected. These perception will be then a kind of secondary goal which will try to become true by the same mean as the primary goal.

This retro-propagation enables therefore a kind of planning by promoting the choice of behaviors contributing to achieve the goal of the ASM. The compulsory nature of some links introduces a natural ordering between these advantaged behaviors : the consequence of one behavior will be necessary to select the following. As these form of planning coexists with direct propagation, unexpected consequences of behaviors, or errors in the sensory system will be immediately taken into account.

The precise amount of activation $\partial\alpha$ propagated through the links are :

- in the *perception P to behavior B link*
  - **direct sense** : $\partial\alpha = \theta_d f_{PB} \alpha_P$
  - **indirect sense** : $\partial\alpha = \theta_i f_{PB} \alpha_B$
- in the *behavior B to perception P link*
  - **direct sense** : $\partial\alpha = \theta_d f_{BP} \alpha_B$
  - **indirect sense** : $\partial\alpha = \theta_i f_{BP} \alpha_P$

where θ_d and θ_i are two coefficients regulating the relative importance of direct and indirect propagation of activation, i.e. the weight of reactivity versus planning. The values of θ_d and θ_i range from 0 to 1, with the constraint $\theta_d + \theta_i = 1$.

As we have seen, components start with a given level of activation, and receive further activation from external sources or through links. As a consequence, activation level can only increase, and it is easy to prove that activation level will diverge in several components. We choose not to introduce a decay to compensate this divergence, but to normalize the activation through the network. The key values are in fact the relative importance of activation levels between the behavior components.

We have described one step of propagation of the activation through the network. The evolution over time of the whole network is synchronous, and based on a discrete time calculation of activation propagation. The concatenation of the several propagation phenomena leads to the following global equations :

$$\begin{cases} \alpha_{P_j}^{t+1} &= \alpha_{P_j}^t + \theta_d \sum_{B_i \in \mathcal{B}} f_{B_i P_j} \alpha_{B_i}^t \\ & \quad + \theta_i \sum_{B_k \in \mathcal{B}} f_{P_j B_k} \alpha_{B_k}^t \\ & \quad + \sum_{G \in \mathcal{G}_{P_j}} \alpha_G + v_{P_i}(s^t) \\ \alpha_{B_j}^{t+1} &= \alpha_{P_j}^t + \theta_d \sum_{P_i \in \mathcal{P}} f_{P_i B_j} \alpha_{P_i}^t \\ & \quad + \theta_i \sum_{P_k \in \mathcal{P}} f_{B_j P_k} \alpha_{P_k}^t \\ & \quad + \sum_{G \in \mathcal{G}_{B_j}} \alpha_G \end{cases}$$

with $\mathcal{G}_B$ et $\mathcal{G}_P$ the set of goals introducing activation in B and P respectively.

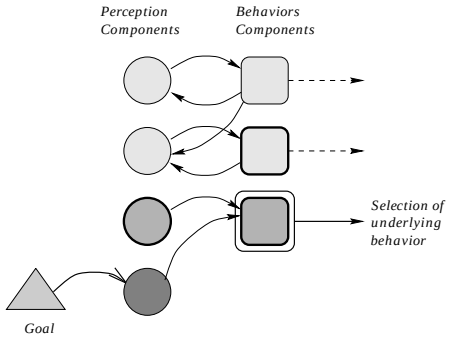

Figure 2: *Graphical synthetic representation of the flat action selection network. Perception and behavior components are placed in two facing columns. Links are represented between the components. The goal is represented by a triangle with a link to its aim. Gray level in components indicates the activation level of it. Bold border for perception component means its truth value is true. Bold border for behavior components indicate all its mandatory conditions are true, and that it is thus choosable by the ASM. A double bold border for a behavior component means this is the chosen behavior of the ASM.*

2.3 Convergence of activation levels and action selection procedure

The convergence of the flat ASM has been proven in a separate work [4]. It is guaranteed by mathematical properties of the equations of evolution of the network, provided the compliance to the range for parameters we put forward in the preceding sections. We suppose that during the convergence time, the values of sensor data[2] is stable. Though the mathematical convergence is theoretically reached at infinite time, empirical results show that just a few time steps are necessary to reach it practically.

[2] or at least the truth value of perception

This convergence enables us to explain the action selection process. When the equilibrium has been reached, we select in the set of all behaviors those whose mandatory conditions are all fulfilled[3]. The distribution of the different activation levels in this list of potential behaviors gives us enough information for choosing a good candidate, usually the most activated. Actually, according to our preceeding explanation, this distribution reflects the compromise between the matching of the behavior with the current situation and its efficiency for reaching the flat ASM goal.

Since the action execution is not instantaneous, another question to be answered is *when* the selection must occur. The truth value of perception is the only information that come to the ASM about the environment. Doing a selection each time one of these truth value change is therefore a good idea.

3 Description of the hierarchical architecture

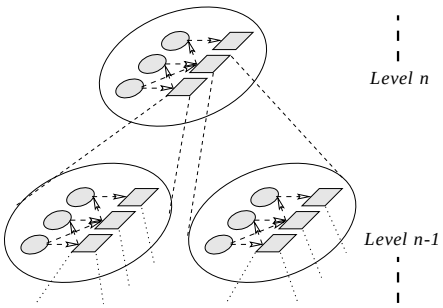

Figure 3: *Introduction of flat ASM in a hierarchical framework to build a global ASM . Behaviors of level n are flat ASMs selecting between behavior components of level* $n-1$.

According to the idea exposed in the introduction, the flat ASM structure that we have described become the construction element of a global hierarchical ASM. Seen from outside, a flat ASM driving several reflexes has a global behavior that tends to fulfill a precise goal. Though this goal is not automatically fulfilled, the use of this ASM will induce several changes on the environment, as would a reflex behavior. Its use is appropriate provided some conditions on the environment, that will increase the probability of its goal fulfillment. Therefore, seen from the outside, this ASM has exactly the same properties as one of its behavior components or as a reflex behavior. So there is no problem of identifying flat ASMs with complex behaviors, and integrate them in a hierarchical ethological-like mechanism.

We can take a constructivist viewpoint, starting from a given set of reflex behaviors. It is possible to build several complex behaviors by grouping some of the reflexes into a flat ASMs, designed to reach a defined goal. We just have to find appropriate perceptions to do in each flat ASMs an efficient action selection. We then will have a set of complex behaviors achieving specific goals. We can group these complex behavior into new flat ASMs, building even more complex behaviors, and so on until all the behaviors are bounded into a single hierarchy. This can be seen as the bottom-up construction of a hierarchical ASM.

The different ASMs of the hierarchy are propagating their activation levels autonomously. Nevertheless, in each ASM, there can be only one sub-behavior that can be selected at one time. Thus, the top ASM will select one of its sub-behaviors B, enabling the further selection of the behaviors that are in the B "branch". All the behaviors that are not in an ASM under B could not be selected at that time. This reasoning is recursive, and there is therefore only one active behavior at one time at each level of the hierarchy. Propagation of activation is therefore not necessary in other flat ASMs of the hierarchy, which constitute an appreciable economy of computing power.

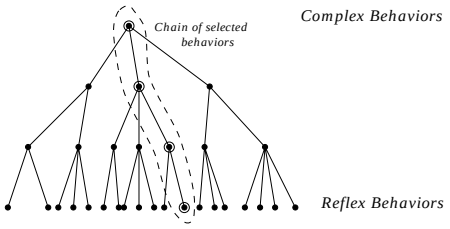

Figure 4: *At each time, there is one behavior selected at each level of the hierarchy. This constitutes a chain from the stand-alone node at the top of the behavior tree, to one of the reflexes at the bottom of it.*

4 Related works

The action selection problem is commonly found in reactive robotics applications, whether treated as a single problem or integrated as a function of a general architecture. The subsumption architecture is an exemple of integrated action selection mechanism ([3],[2]), though the selection is based on a hard-wired precedence order of behavior "levels". The same kind of hard-wired selection can be found in Steels' work ([19],[20]), with a formalism of motivation functions. In all these approaches, goals are preset, as well as reactions to events, and changing them requires a reprogramming of the agent.

Some recent work of Michaud *et al.* has proposed some way of combining reactivity and planning [16]. The way planning is achieved is there very different from our approach, because it is made at a symbolical level of world representation, and not at the basic action planning level. The agent constructs in this approach a topological graph representation of its environment by exploring it systematically. Once this representation is learned, the robot chooses a path in this graph by basic graph exploration and uses this topological path to implement motivations at its reactive level. Thus, after a reactive exploration phase, the system becomes a planner of reactive rules, which is rather different to implementing both planning and reactivity simultaneously.

Donnart and Meyer describe in [5] an autonomous agent architecture based on the hierarchical organization of learning classifier systems. However, it is a hierarchy of functions : planning, reactivity, learning, and influencing choices. The

[3] i.e. all the source perceptions of mandatory links whom target is the considered behavior must be true.

cornerstone of this approach is the choice of a reward function that favors one rule of action over one other in a competition between rules. If we make a comparison with our approach, our flat ASM finds this reward function and adapt it each time there is a change in the environment. Moreover, planning and reactivity are separated, which induces the creation of an explicit plan, whereas our planning method directly influences the action selection without the need of a plan.

Rosenblatt and Payton introduced a hierarchical action selection mechanism in [17]. Their solution is loosely inspired on the principle of subsumption architecture, and relies of a hierarchy of neuron-like nodes that take their input from upper level nodes, and both proprio- and exteroceptive sensors. This architecture has not been implemented by the author, and the action selection is rather hard-wired by the nodes connection strengths which determine the input/output function of the architecture. There is no real selection, but rather a combination of influences on the effector units. Furthermore, there is no kind of planning in Rosenblatt's architecture .

5 Experimentations

To test the validity of the ideas introduced in our action selection architecture, we conducted a series of experimentations on a simulated autonomous robot.

5.1 Experimental setting

We used a simulator of the Khepera robot to make these experiences. The Khepera robot is a small robot platform specially suited for indoor laboratory experimentations. It is cylindrical, about 3 inches in diameter and an inch tall. It has two symmetrical wheels commanded in speed by two independent motors, what enables it to turn without changing its position. It has in its basic configuration eight infra-red sensors that enable him to detect obstacles in active mode and detect light in passive mode. The principal characteristic of it is that it is low-cost: sensors are not identical nor linear nor well calibrated. Both motors and sensors are noisy. Thus the control architecture of such a robot has to be fault tolerant. The simulator of Khepera robot has been written by Olivier Michel, and constitute a realistic approximation of it.

We used several reflexes as a basis for experimentation. We inspired from Braitenberg to build a two-neuron reflex. Each neuron receives values from the eight IR sensors, and delivers a motor speed command to one of the motor. By tuning the weights of the different links between sensors and neurons, we managed to build several reflexes we arbitrarily named according to their observed properties as presented in table 1. On top of that, we add a gradient following behavior in the simulation so that the robot can be attracted by some specific point in the environment. This can be thought of as a radio beacon emitting a recognizable signal for the robot. It has just a directional indication on this radio signal. There can be several distinct beacons in the environment. We denote the corresponding behavior *go-to-M* where M is the name of the beacon.

We used several perceptions based essentially on thresholding some of the sensor values. This enables us to build some obstacle detector, whether omni-directional or directional, some radio-signal perception to indicate if the robot is very close to the radio-beacon and a perception to detect when the robot don't move any longer.

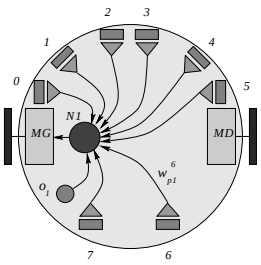

Figure 5: *The Khepera robot with a reflex architecture. The two motors (MG and MD) and the eight sensors (numbered from 0 to 7) are represented. One of the two symmetrical neuron is represented (N1). It receives input from all the sensors with a weight w^n_{pk} (p for proximity, k is the neuron number, n is the IR sensor number), and a constant input o_k. The notations are used to define several different reflexes in table 1.*

Once perception and behavior components are available, designing the links is done by establishing conditions of behavior use, and expected consequences of the use of a behavior. For each perception P conditioning the choice of B, we put a link from P to B of an arbitrary strength, let say 1. The same method is applied for choosing the strengths of consequence links. If we think that one condition is more important than another, we can lower the strength of the weakest one. If we have an idea of the probability of P becoming true after the use of B, we can use it as the strength f_{BP}. Anyway, the design of link strengths proved to be very robust, and no problem have been observed with this rather brutal method of link design. The strengths of the links can thereafter be adapted by the statistical estimation process exposed in [15], and by reinforcement learning; but this go beyond our current topic.

We conducted two sets of experiences :

- The first one was a mere testing of the reactive planning properties of the flat ASM. The global ASM was just a flat ASM containing an obstacle avoidance and go-to-A behavior, and perceptions indicating if there is or not an obstacle, and if the robot is close or far of the beacon A (cf. figure 6). The goal was to make the perception *near A* becoming true. We tested this ASM in several configurations of the initial position of the robot, of beacon A position and in several environments.

- The second experience aimed to illustrate the goal combination property of the hierarchical architecture. We compared two ASMs with the same two goals : *near A* and *near B* required to be fulfilled simultaneously. The first ASM was a flat one combining in a manner similar to the first experience obstacle avoidance, go-to-A and go-to-B behavior, with the appropriate perceptions (cf. figure 7). The two goals were introduced in it directly. The second ASM was doted with a hierarchical organization separating the two goals in two flat ASMs of lower level (cf. figure 8). These two ASMs where enclosed in a upper level flat ASM, constituting a very simple hierarchy. The upper level has just perception components able to determine

Behavior	w_{p1}^0	w_{p1}^1	w_{p1}^2	w_{p1}^3	w_{p1}^4	w_{p1}^5	w_{p1}^6	w_{p1}^7	o_1	o_2
Obstacle avoidance	0	0	0	1	1	1	0	0	0.5	0.5
Left wall following	0	0	0	0	1	1	0	0	0.5	0.6
Right wall following	0	0	0	0	1	1	0	0	0.6	0.5
Corridor following	1	0.5	-1	-1	-0.5	1	1	1	0	0
Left static turn	0	0	0	0	0	0	0	0	-1	1
Right static turn	0	0	0	0	0	0	0	0	1	-1
Forward move	0	0	0	0	0	0	0	0	1	1

Table 1: *Coefficients for Khepera reflexes. The second neuron has the same coefficients excepted for the constant input o_2.*

whether the goals of the two lower ASMs are or are not fulfilled. We chose a stochastic choice procedure at the upper level after activation propagation to enable the global architecture to alternatively fulfill one goal and the other.

5.2 Results

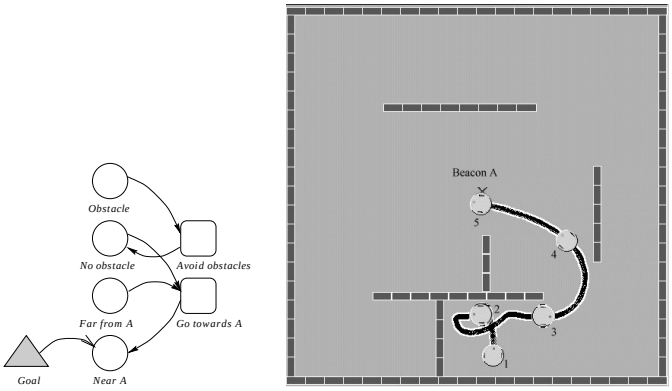

Figure 6: *Trajectory of the robot in experience 1. The dark line represents the trajectory of the robot over time. Its starting position is at (1). At (2) it encounters an obstacle and deviates to its left. Reaching another wall, it changes its behavior to go towards A. Reaching again the wall, it alternatively avoids it and goes towards A, resulting in a following of the wall. At the end of the wall, at (3), it go freely to the beacon passing (4) and reaching it at (5).*

We tested these ASMs architectures in a large number of initial configurations, goal position, and in different environments. We did not compared in the same conditions our ASM with other ones, as did Tyrrell in [22]. There is in fact no simple way of evaluating numerically the performance of an ASM. A case by case study has been done, providing homogeneous qualitative results that are illustrated here by representative examples.

The first experience shows, as illustrated by figure 6 that the global behavior of the flat ASM was very efficient. Some tuning on the links strengths was still necessary to get a good predominance of obstacle avoidance in the right circumstances, but it was very easy. When the proper tuning has been discovered, the robot manage to reach the beacon nearly anytime. The only problem we have noticed happened when the robot followed a straight path to the beacon and entered a cul-de-sac. As it had no global representation or map of its environment, it could not anticipate the problem. It got stuck into it, avoiding the bottom, and then

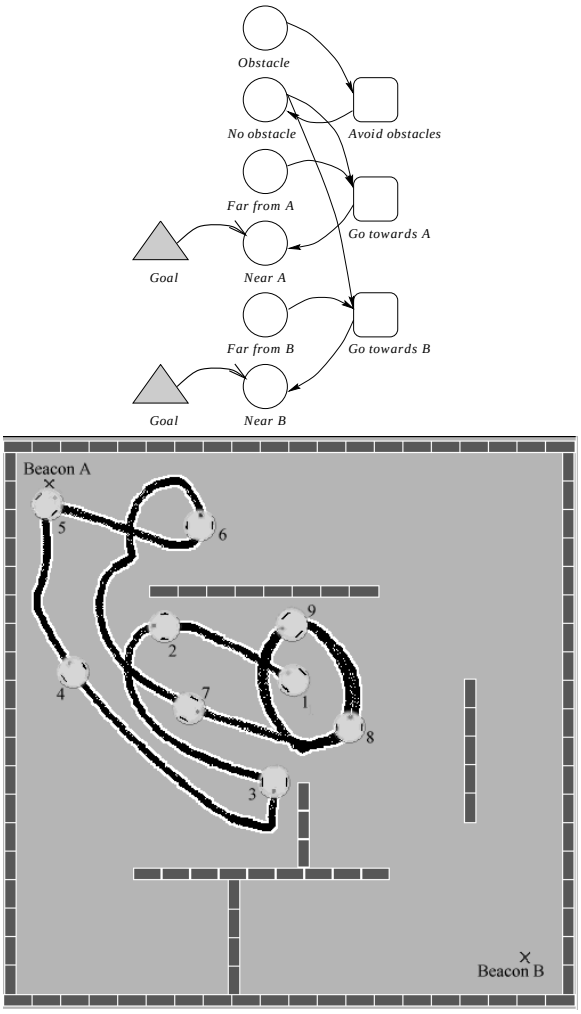

Figure 7: *Trajectory in experience 2 with a flat architecture . The robot has two simultaneous goals in the same flat ASM:* go to point A *and* go to point B. *At first, the robot is going towards A. Then, it changes at (2) because he must choose a new behavior to avoid the wall. Once the wall is avoided, it decides to go towards B. Then, its changes back to A at (3) for the same reason. At (5), it reaches beacon A and decides to go to B, but at (6), it changes after avoiding the wall, and then change back again. Finally, the robot circles at the center of the environment at (8) and (9) in a very stable deadlock. This illustrates Tyrrell's critic on Maes-like architectures.*

going back to it, attracted by the beacon. This problem has been easily solved by replacing obstacle avoidance by a wall following behavior.

The second experience enabled us to rediscover the problem of Maes' ASM as stated by Tyrrell in its symbolic environment simulation [23]. Figure 7, presenting an example of trajectory produced by the first tested architecture, shows clearly what was happening. The robot never managed to fulfill one of its two goals. Each time the robot perceived an obstacle, an action selection occurred. One of the two behaviors enabling to reach a beacon became stronger than the other alternatively. The robot therefore changed his mind at each action selection, circling most of the time in the middle of the arena and reaching very seldom one of its goals. The second tested architecture was far more efficient. Figure 8 shows an example trajectory of the robot with this hierarchical ASM. The hierarchy enabled it to stick to one goal once it has been chosen, as long as it was not fulfilled. This stabilization in goal choice enabled the alternative fulfillment of both of them.

6 Discussion and future work

From the preceeding principle example, we can say that we fulfilled the requirements we postulated at first on at least three points. Maes' mechanism enables effectively to tune between reactivity and planning, and its integration in a hierarchical organization works fine and manage to deal with conflicting goals. These good properties have yet to prove their scalability to a real and complex robotics application.

The fourth requirement – adaptivity – can be said to be partially achieved. Maes' ASM is in fact able to *adapt* a certain number of unexpected events : a misleading perception truth value, an unexpected consequence of an action or a dynamic change in the goal. With the help of statistical estimation of behavior consequences, it is also possible to evaluate the probability that one perception become true after the use of a given behavior. The strengths of the links can be set to the value of this probability, as stated in [15]. It enables on line learning of behavior to perception links, even with initial random values. We tested it in an other work to be published later with good results. A way to make on line adaptation on perception to behavior links would be to use reinforcement techniques with a reward function associated to each goal in flat ASMs. This is the object of our current work.

Much more can be done in the field of adaptation. One long term aim could be to find a method to adapt the organization of the whole ASM : which perception to be put in each flat ASM, how can reflex behaviors be grouped in bottom flat ASMs, how can the structure of the hierarchy be adapted and when, etc. This is obviously a wide and difficult program.

This leads us to point out another current limitation of our approach. We exposed a two-level hierarchy in our tests. If building a hierarchical organization with more levels poses no theoretical problem, it induces practical ones. Finding the right perceptions for higher level flat ASMs is not an easy problem. Intuition suggests that the perceptions and the behaviors at higher levels will become more and more abstract and symbolic. This implies that we risk the apparition of the *symbol grounding problem* [9] that was avoided until now thanks to our bottom up situated approach [18]. It will be probably necessary to integrate perception components computed on the basis of complex environment representations such as topological, geometric or feature maps, or

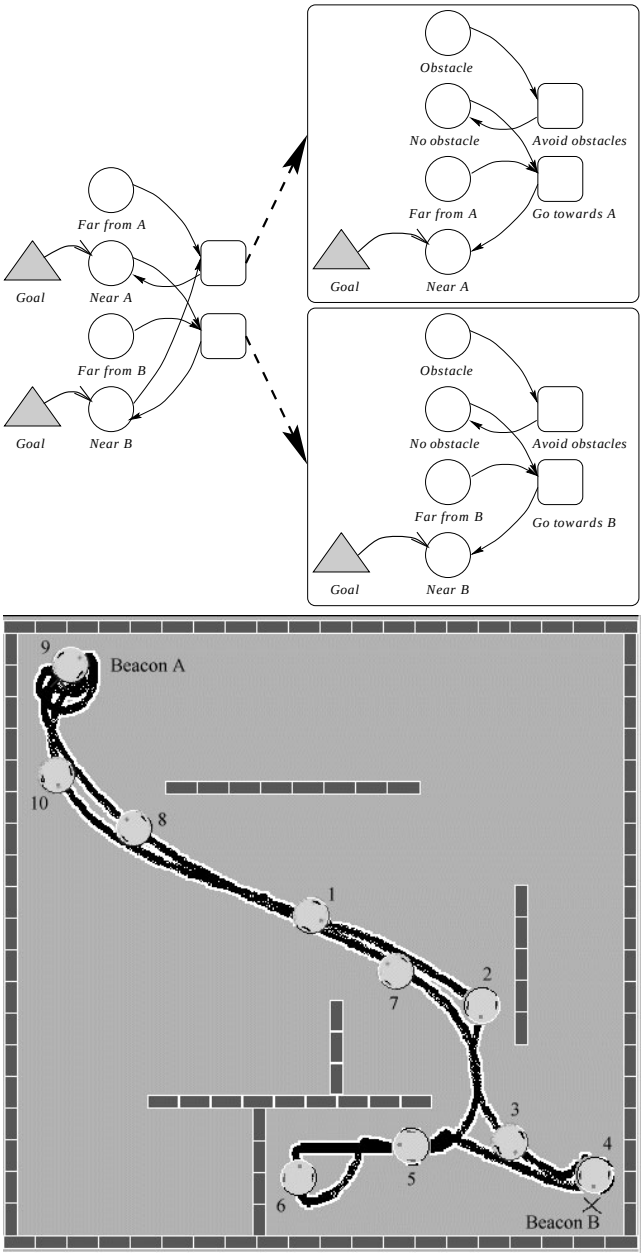

Figure 8: *Trajectory in the two-level hierarchical ASM with the same goals as in the preceeding experience. The robot starts at (1) and goes to B (2,3 and 4). Then it decides to go to A (5 to 9). On its path to A, it enters a cul-de-sac (5 and 6). At the exit of this cul-de-sac, the robot has not changed its goal (7). Finally, the robot goes back towards B (10). This shows the stabilizing effect of separating the different goals in separate flat ASMs: the robot manage to fulfill both goals successively, even if sometimes problems like entering a cul-de-sac occur.*

symbolic models, whose manipulation is demanding in computing power. This problem is also linked to the increasing time scales of flat ASMs evolution as we go up in the hierarchy. All these points are currently under study, but could not get a real answer without a more complex experiment.

This is why our future work will be to implement the action selection architecture on real and complex robots of different kind. Our first work will be on a car-like platform with vision-based perception and reflexes, and long term goal is to use an underwater robot. These real world implementations will feed our conceptual work along three parallel directions :

- *adaptation* of the action selection, including the issues of putting reinforcement learning, modifying the hierarchy structure, finding new perceptions and behaviors, etc.
- managing the *real time* problem : scheduling and allocating computer resources to perception computation, flat ASMs, reflexes execution, etc.
- *integration* of robotics components not specifically designed for our action selection mechanism : encapsulation of automation control laws, integration of complex world models, etc.

References

[1] G. Baerends. The functional organisation of behaviour. *Animal Behaviour*, 24:726–735, 1976.

[2] R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, 3 1986.

[3] Connell. A colony architecture for a robot. Technical report, MIT AI Lab, 1986.

[4] V. Decugis and B. Beauzamy. Convergence of a reactive planning algorithm. *submitted to Journal of Applicable Analysis*, 1998.

[5] J.Y. Donnart and Meyer J.A. A hierarchical classifier system implementing a motivationally autonomous robot. In D. Cliff, P. Husbands, J.A. Meyer, and S.W. Wilson, editors, *From Animal to Animat III, Third International Conference on Adaptive Behavior*, pages 144–153, 1994.

[6] M. Drummond. Situated control rules. In R.J. Brachman and H.J. Levesque, editors, *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 103–113, San Mateo, CA, USA, may 1989. Morgan-Kauffmann.

[7] M. Drummond and J. Bresina. Anytime synthetic projection: maximizing the probability of goal satisfaction. In *AAAI90 Proceedings - Eight International Conference on Artificial Intelligence*, volume 1, pages 138–144, Cambridge, MA, USA, july 1990. MIT Press.

[8] M. Drummond, K. Swanson, J. Bresina, and R. Levinson. Reaction-first search. In R. Bajcsy, editor, *Proceedings of International Joint Conference on Artificial Intelligence*, volume 2, pages 1408–1414, San Mateo, CA, USA, august 1993. Morgan Kauffmann.

[9] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[10] K. Kanazawa and T. Dean. A model for projection and action. In N.S. Sridharan, editor, *IJCAI 89 Proceedings of the eleventh International Joint Conference on Artificial Intelligence*, volume 1, pages 985–990, Palo Alto, CA, USA, august 1989. Morgan Kauffmann.

[11] R.E. Korf. Real-time heuristic search. *Articial Intelligence*, 42(2-3):189–211, march 1990.

[12] K. Lorenz. The comparative method in studying innate behavior patterns. *Symposia of the society of Experimental Biology*, 4:221–268, 1950.

[13] P. Maes. How to do the right thing. *Connection Science Journal*, 1(3), February 1990.

[14] P. Maes. Situated agents can have goals. *Robotics and Autonomous Systems*, 6:49–70, 1990.

[15] P. Maes. Learning behavior networks from experience. In F. Varela and P. Bourgine, editors, *Towards a Practice of Autonomous Systems*, pages 48–57, 1991.

[16] F. Michaud, G. Lachiver, and C.T.L. Dinh. A new control architecture combining reactivity, planning, deliberation and motivation for situated autonomous agent. In *From Animal to Animat IV*, pages 245–254, 1996.

[17] J.K. Rosenblatt and D.W. Payton. A fine-grained alternative to the subsumption architecture for mobile robot control, 1988.

[18] S.J. Rosenschein and L.P. Kaelbling. A situated view of representation and control. *Artificial Intelligence*, 73, 1995.

[19] L. Steels. Mathematical analysis of behavior systems. In *Proceedings of the Perarc Conference*, Lausanne, 1994.

[20] L. Steels. Intelligence - dynamics and representations. In L. Steels, editor, *The Biology and Technology of Intelligent Autonomous Agents*. Springer-Verlag, Berlin, 1995.

[21] N. Tinbergen. The hierarchical organization of mechanisms underlying instinctive behaviour. *Experimental Biology*, 4:305–312, 1950.

[22] T. Tyrrell. *Computational Mechanisms for Action Selection*. PhD thesis, University of Edinburgh, 1993.

[23] T. Tyrrell. An evaluation of Maes's bottom-up mechanism for action selection. *Adaptive Behavior*, 2(4):307–348, 1994.