Hierarchy of behaviours

Application to the homing problem in indoor environment

Nicolas Mansard, Olivier Aycard, Carla Koike
INRIA Rhone-Alpes - GRAVIR Laboratory
Campus de Montbonnaut, 655 av Europe, 38330 Montbonnot Saint Martin, France
E-Mail : `Firstname.Lastname@inria.fr`

***Abstract*— Living beings are often observed switching strategies in response to a changing environment. However, autonomous robotics mostly implements a single behaviour well suited to a particular task such as navigation, localization and so on. Actually, one burning issue of autonomous robotics is to manage a complex task starting from a set of simple behaviours. In other words, the robot has to choose the optimal behaviour given the sensori-motor context in order to build a global and coherent process. This is usually done by a strict specification from the programmer. In this article, we put forward a framework called behaviours hierarchy that handle elementary ability to respond to a given task. We show that this framework leads to the continuous application of an adequate behaviour depending on the environment. Finally, we propose a general method to implement this framework using Bayesian programming.**

I. Introduction

We propose in this article a general approach to structure a set of behaviors. This builds a global relevant strategy, that allows our robot to reach a goal, which is not accessible to each behavior alone. We call this structure hierarchy of behaviors. Contrary to the subsumption architecture, given in [2], our structure is very intuitive. A simple but general behavior is put on the base of the hierarchy. Higher you go in the hierarchy, more specific but more powerful behaviors you get. Each behavior is thus specialized in a group of sub-behaviors, that are more restricted, but more efficient on their domain. At each time of the execution, the set of the state variables is used to decide which behavior to use. In this way, our structure can be understood like a finite state automat.

We have used this approach to solve the homing problem in indoor environment. Our robot works in total autonomy, realizing various tasks we don't take care here. During his trip, it records knowledge that will be used to go back home. When his batteries' level of charge is below a fixed threshold, it begins the homing task. We have chosen a global strategy that consists in putting red bolt on the floor at each junction. Then, during the return journey, the robot has to choose at a junction the marked direction. This very simple strategy has been chosen to validate and illustrate with a relevant experience the use of our hierarchy of behavior to get complicate strategy using only simple behaviors.

We will begin our presentation by putting in evidence what in the state of art of our problem justify our choice. This part will also explain the intuitive ideas of our structure, and quickly expose what we expected to do when we have begun our research in this way. Section III will present our application. We have chosen to begin by this way to give a clear and simple illustration to refer to, when reading the following of this article. We give then the mathematical formalization of our structure (section IV) and justify the relevance of all these definitions by a second set of definitions and theorems (section V). A direct method is presented in very short terms that implements the hierarchy of behaviors, using Bayesian robot programming (section VI). Finally, we present the results obtained when testing the robot implementation in real condition (section VII).

II. State of art

A. Classical approaches

A lot of different ways of research have already been proposed for the navigation problem. The classical approaches use a set of knowledge (a map) to localize the robot at each step of the process. These methods are usually divided in two groups, according to the kind of map they need.

The metric maps represent the environment with geometrical features. A first commun way to build the map is the occupancy grid [5]. The environment is discretized and each cell of the grid contains the probability to be occupied by and obstacle. A second way is to modelize the environment by a set of geometric features [4], [3]. These methods provide exhaustive maps. Each detail is stored in the map of the environment. When computing a path or tracking a trajectory, algorithms must deal with this abundance of information.

On the opposite, topologic maps are semanticaly richer. Acquired informations are treated immediately and stored in a way as usefull as possible to further corresponding actions. A common way of proceding is to build a graph. The significant places are nodes of the structure. They are linked by edges representing paths between the places. Labels are then added to the graph to store all the required knowledges about the environment. These models are lighter: only usefull informations are stored. But it's also limited by the semantic chosen. Data extraction is also more problematic.

B. Hybrid approaches

An exhaustive comparison of these two classes of dealing with the environment is done in [12]. The author gives as a conclusion an interesting summary of their qualities and lacks. Of course, the first conclusion is that the two methods are complementary. Studies comparing the two methods tend to prove that the *good way* of proceed is to use mixed algorithms. This was already proposed in [3]. But really mixed methods have been proposed recently. In [9], Thrun proposed

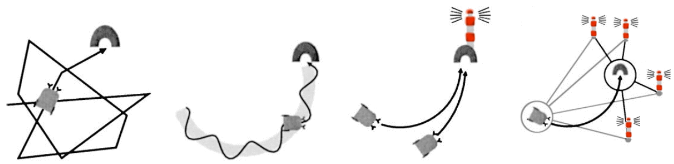

Fig. 1. The four local navigation methods

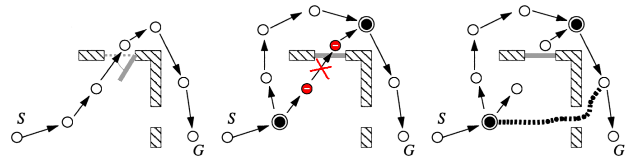

Fig. 2. The three navigation methods using knoledge about the environment

to build a simple topologic map (finite state automata) from a dynamic occupancy grid. The topologic map was used to quickly planify the robot trajectoy.On the contrary, Kuypers *et al.* proposed to enrich their topologic map until they obtain a metric map of the robot environment [7]. A third method has been proposed in [10]: a global topologic map is used. In the nodes of the graph, local places were documented by an occupancy grid. These metric maps were used to obtain a very good precision (about 1cm) at the end of the robot's trip.

These three examples of mixed methods are really interesting and provide extremely good results. But they are constructed for specific purposes and environments. We propose in this article a general theory that will at end permit to build general mixed methods by a rigourous structure.

C. Biomimetic approaches

Our research of a structure to arrange navigation algorithms has been based on biomimetics observation. The bio-inspired robotic copies the algorithms proposed by the biologist to explain biologic behaviours. These methods are generally simple but efficient strategies, offering very good results in specific situation and a classification as been proposed in [6]. It is build as a hierarchy of behaviours, and numerous indexes about robotic implementations are given. The navigation algorithms are divided in two groups, depending on whether they use a map (path searching), or not (local navigation). The first group are very simple algorithms. The authors distinguish four classes, from the simplest to the most complex one: random search, path tracking and odometry, beackon's tracking and guidance from numerous significant marks. The four behaviours are represented on Fig. 1. This first hierarchy is very interesting since the authors propose a continuation in the domain generally used in the robotic. The algorithms using maps are segmented in three classes. When applying the simplest method, the robot associates a place it knows with a specific action to apply. This only permits to follow a linear road. Using the above class, topologic guidance, the robot is now able to compare several path he knows from a way to another, and to select the more relevant one. The last algorithm, named survey navigation, adds the faculty to compute new path in unknow territory. It is very similar to metric maps. Fig. 2 representes these three behaviours.

D. Synthesis

The classical robotics propose complet methods. In particulary, using mixed algorithms, robots are able to acheave complexe tasks. But we can wonder if using the most achieved methods of the biomimetic hierarchy is relevant. The behaviour will be correct, but is it requisite to use the whole compute capacities to a task that requires less ? The works of bioinspired roboticians are often appropriate for only very specific conditions. But they propose above all a hierarchy of guidance methods that can be used as a base to generalise the fusion of several algorithms in a mixed one. We have presented these work, not as a list of minimal strategies examples, but to introduce the notion of hierarchy. In our opinion, this way of regard a behaviour directly provides a general and rigourous structure to mixed guidance algorithms.

In the following, we will proposed a formalisation of these concepts. A first implementation will be then presented as an example of how all this can by applied. By combining very simple guidance algorithms, we obtained a good global behaviour, that makes our robot able to find its path back to the home. These simple behaviours are cheap in terms of computing ressources and *a priori* knowledges. This first implementation valids the usability and interest of our structure.

III. A SIMPLE STRATEGY FOR HOMING

For didactic reasons we will start by presenting the specific implementation before going on to expand on the general theory. However it is important to keep in mind that this simple experiment is only for validation purposes.

We limit our guidance algorithm to solve the homing problem in an indoor environment. Since it a just an experimentation to valid de following theory, this limitation is not a problem: we don't want to keep any generality for these first tests. As other guidance algorithm, this is divided in two part: the construction of the map first. And the way this map is used to achive the home return. To stay in the biomimetic domain, our map is physically build by puting red token on the floor at each corridor crossing. The global strategy is thus to simply follow the current corridor, and to choose at each crossing the path pointed by the biggest number of tokens. The home is indicated by more than ten token on the floor.

During its travel away from home, the robot tracks corridor crossings. A token is put off when the distances mesured right and left (+ and $-90°$) rise suddently. This very simple crtierion can leads the robot to put off tokens in places that aren't crossing. But these mistakes will not represent problems during its way back.

First, the robot detects the corridor around it, and follow the corridor direction. When a crossing is detected, the robot executes of $360°$ rotation to count the number of token around it. It then chooses the path pointed by the maximum number of tokens and rotates to follow this direction. If one of the

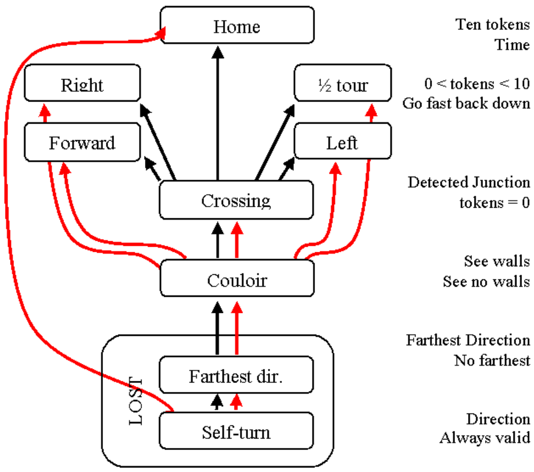

Fig. 3. Hierarchy of behaviours used in the experiments. On the right, the criteria to select the behaviour to be adopted are explicited. When the first one is true, the robot choose a higher behaviour in the hierarchy. A lowest behaviour is applied when the second criteria becomes true.

directions is marked by more than ten tokens, a specific behaviour is proceeded to enter the home. Finally, two last behaviours are added to deal with unknown situations. The first one simply drives the robot along the farest direction detected. This is useful to make the robot run off a room, or from other places that can't be recognized as corridors. The last behaviour only makes the robot rotate around itself, when the farest distance is not so farest. Because the robot can only mesure distances on the $180°$ behind it, this specific behaviour is necessary to keep the robot away from dead-end.

Now that the global and local strategies have been decided, the criteria to decide how and when to pass from a behaviour from another have to be explicited. This is presented has a finit state automata (see Fig. 3. Two different types of edges between behaviours are used: the first one is followed to go higher in the behaviour hierarchy, when a more specific behaviour is required. For instance, if the robot is applying the path-following behaviour, and detects a crossing, the criterion of the above behaviour *crossing-detected* becomes true, and this behaviour is then applied. A second criterion is used to specify that the current specific behaviour is no more valid. The robot adopts then the below and more generic behaviour.

IV. Formalization

In last section, the global strategy to combine simple behaviours has been detailed by an intutive way. Now, these intuitive ideas and notions will be explained in a more formal way, but keeping in mind the intuitive application to explain and justify our mathematic choices. First, we will need a definition of what a behaviour is. Then a definition of a hierarchy of behaviour will be provided, and some proofs about the robot's global behaviour will be given.

A. Sensory-Motor space

First of all, the definition of a sensory-motor space is given. This definition was already given in [1] and it will be applied in the notion of behaviour presented in the following.

Definition: Sensory-Motor space

The **sensory-motor variables** are a couple $(\mathbf{V_S}, \mathbf{V_M})$ of sets of variables such that

$$\mathbf{V_S} \cap \mathbf{V_M} = \emptyset \tag{1}$$

Let N_S be the cardinal of the set $\mathbf{V_S}$, and N_M the cardinal of $\mathbf{V_M}$. Let N be the number of variables:

$$N = N_S + N_M \tag{2}$$

The set $\mathbf{S}$ of possible values for the variable of $\mathbf{V_S}$ is called **sensor space**. The set $\mathbf{M}$ of possible values for the variable of $\mathbf{V_M}$ is called **motor space**. The sensory-motor space is

$$E = \mathbf{S} \times \mathbf{M} \subset \mathbb{R}^N \tag{3}$$

B. Behaviour

Definition: Behaviour

A **behaviour** on a sensory-motor space $\mathbf{S} \times \mathbf{M}$ is a triple

$$(\mathbf{f}, \mathbf{C}, \mathbf{I}) \in \mathcal{P}(\mathbf{S} \times \mathbf{M}) \times (\mathbf{S} \to \{0,1\}) \times (\mathbf{S} \to \{0,1\} \times \mathbb{R}^+) \tag{4}$$

$\mathbf{f}$ is a subset of the behaviour. This can be seen as a partial function that associates a motor answer (vector of $\mathbf{M}$) to each sensor input (vector of $\mathbf{S}$). $\mathbf{f}$ is called the strategy associated to the behaviour $(\mathbf{f}, \mathbf{C}, \mathbf{I})$. $\mathbf{C}$ and $\mathbf{I}$ are the criteria which decides wich behaviour should be applied. $\mathbf{C}$ is called the credibility value of the behaviour. $\mathbf{I}$ is called the relevance value of the behaviour.

Interpretation of the definition

At each step of time where the robot should take a decision about what to do, $\mathbf{C}$ is a boolean depending on what the robot sees. The credibility is a criteria that can be used to decide if the current behaviour is valid. The relevance is a criteria that helps to decide if the behaviours situated above in the hierarchy are usefull. $\mathbf{I}$ is a couple of value. The first one is a boolean. The second one is considered only if the first one is true. Intuitively, this second value is a criteria to decide which of the two behaviours is more usefull: the next behaviour chosen will be the one with the higher second value.

C. Hierarchy of behaviours

Definition: hierarchy of behaviours

A hierarchy of behaviours is a set of behaviours $\mathbf{B}$ and a relation $\mathbf{R} = (\mathbf{R}_-, \mathbf{R}_+)$ so that:

- $\mathbf{R}_-$ and $\mathbf{R}_+$ are two relations on $\mathbf{B}$ and:

$$\begin{aligned} &\forall (b_1, b_2) \in \mathbf{B}^2, (b_1 \mathbf{R} b_2) \\ &\quad \Leftrightarrow (b_1 \mathbf{R}_- b_2) \text{ OR } (b_1 \mathbf{R}_+ b_2) \end{aligned} \tag{5}$$

- $\mathbf{R}$ is antisymetric:

$$\forall (b_1, b_2) \in \mathbf{B}^2, (b_1 \mathbf{R} b_2) \Rightarrow \neg(b_2 \mathbf{R} b_1) \tag{6}$$

- $\mathbf{R}$ is acyclic:

$$\begin{aligned} &\forall (b, b') \in \mathbf{B}^2, \text{ if } \exists (b_1 \ldots b_N) \in \mathbf{B}^N \\ &\text{such that } b_1 \mathbf{R} b_2 \quad \ldots \quad b_{N-1} \mathbf{R} b_N, \quad b \mathbf{R} b_1 \\ &\qquad \text{and } b_N \mathbf{R} b' \\ &\qquad \text{then } \neg(b' \mathbf{R} b) \end{aligned} \tag{7}$$

- $\mathbf{R}_-$ is connected

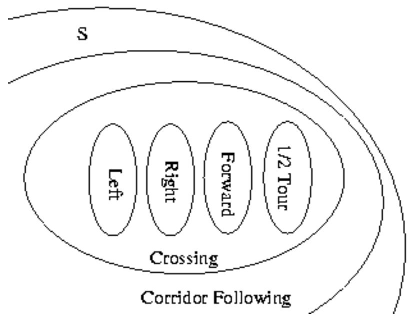

Fig. 4. Tree structure of the sensory-motor space $\mathbf{S}$ definied by the hierarchy presented in section III

- Each behaviour has only one previous behaviour for relation $\mathbf{R}_-$

$$\forall b \in \mathbf{B}, \text{ if } \exists b' \in \mathbf{B} \text{ such that } b'\mathbf{R}_- b, \quad \text{then } \exists! b'' \in \mathbf{B} \text{ such that } (b''\mathbf{R}_- b) \tag{8}$$

Interpretation of the definition

A hierarchy of behaviour can be explained with a oriented graph, where the vertices are the behaviours, and the edge the representation of the relation $\mathbf{R}$. This graph is acyclic from (7). Two vertexes are linked by at most one link (6). A vertex has at most one predecessor for the relation $\mathbf{R}_-$ (8). The hierarchy is defined by two relations:

- The first relation $\mathbf{R}_+$ is called superiority relation. For each behaviour b, it defines the above behaviours, that can receive priority over b (if their relevance value $\mathbf{I}$ become true).
- The second relation $\mathbf{R}_-$ is called inferiority relation. For each behaviour b, it defines which below behaviour should be applied if b becomes inaccurate (if its credibility value becomes false).

D. Tree-structure of the sensory-motor space

The last definition is the formalization of the global strategy defined in the section III. More generally, it is a formalisation of the hierarchy notion, definied in the state of art section. A frequently asked question concerns the relevance to allow only one below behaviour in the definition (8). We understand that this is justified if the behaviours are considered as subset of the sensory-motor space. Each behaviour is then a subset of the below behaviour. A tree structure of the sensory-motor space is obtained, by including subset of $\mathbf{S}$, as shown fig. 4. To select one of the behaviour is equivalent to concider a point of the sensory-motor space. When moving in the sensory-motor space, a behaviour above in the hierarchy is selected if passing in a more specialized subset. On the opposite, if the current behaviour has is credibility value set to false, it is equivalent to leave the current subset and to arrive in the including subset. To allow only one above behaviour in the hierarchy seems thus logical.

V. Using these definitions

We now want to proof that the strategy selected is valid, that is to say will always choose a behaviour of which the credibility value is true.

A. Minimal behaviour

Definition: minimal behaviour

Let $H = (\mathbf{B}, (\mathbf{R}_+, \mathbf{R}_-))$ be a hierarchy of behaviours. A minimal behaviour b_{min} of H is a behaviour with no behaviour below it in the hierarchy, that is to say such that:

$$\forall b \in \mathbf{B}, \neg(b\mathbf{R}_- b_{min}) \tag{9}$$

Corollary: existence and unicity

A hierarchy of behaviours H has an unique minimal behaviour, designed by $\mathbf{B_{min}}(H)$.

Interpretation of the definition

$\mathbf{B_{min}}(H)$ is equivalent to the whole sensory-motor space. This behaviour always provides a motor answer, whatever the sensor inputs are. In the hierarchy defined in section III, the minimal behaviour is *self-turn*.

B. Accessibility

Definition: accessibility

Let $H = (\mathbf{B}, \mathbf{R})$ be a hierarchy, and let b_0 and b_f be two of these behaviours. b_f is accessible from b_0 for an input $s \in \mathbf{S}$ if there exists a string of behaviours $b_1 \dots b_N$ such that:

$$\begin{array}{l} \bullet b_N = b_f \\ \bullet \forall i \in \{0 \dots N-1\} \\ \quad \circ b_{i+1}\mathbf{R}- b_i \text{ and } \mathbf{C}(b_i) = 0 \\ \quad \circ \textbf{ OR } b_i\mathbf{R}+ b_{i+1} \text{ and } \mathbf{I}(b_{i+1}) = (TRUE, m) \text{ with} \\ m = \textbf{max}\{x \text{ tq } b_i\mathbf{R}+b \text{ and } \mathbf{I}(b) = (TRUE, x)\} \end{array} \tag{10}$$

Interpretation of the definition

The accessibility notion is a formalization of which behaviour should be chosen at the current time. There can be at most two possibilities: to go up in the hierarchy, that is to say to use a more specifiedalized behaviour. Or to go down in the hierarchy if the current behaviour has its credibility value to FALSE. In the experimentations, we have always decided to use the more complex behaviour if there was a choice.

C. Coherence

Definition: coherence

A hierarchy of behaviours is coherent if for all current behaviour b, and for all input values $s \in \mathbf{S}$, there is a behaviour b', accessible from b, whose credibility value is TRUE.

Corollary

Let H be a hierarchy of behaviour and let $\mathbf{B_{min}}(H)$ be its minimal behaviour. If $\mathbf{B_{min}}(H)$ is always valid ($\forall s \in \mathbf{S}, \mathbf{C}(\mathbf{B_{min}}(H)) = 1$), then H is coherent.

Proof

It is only necessary to proof that the minimal behaviour is always accessible. Let b_0 be the current behaviour. We will build a string of behaviours from b_0 to a valid behaviour b_f. We add successively some behaviours to the string. If the considered behaviour has its credibility value to TRUE, then we can end the string: we have found a valid behaviour to apply. If not, we can go down in the hierarchy. Because $\mathbf{R}_-$ is connected and tree-like (8), the string will finally ends by finding the minimal behaviour. In both case, we can found an accessible behaviour from b_0.

Interpretation of the definition

This result is very intuitive. It simply proofs that even if the only valid behaviour is the minimal one, the robot can keep a valid behaviour by simply going down in the hierarchy to find an appropriate answer to the current input values.

D. Conclusion

We have formalized the notion of hierachy introduced in the beginning of this article. We have explained how to rigorously link a set of behaviours, by some numerical criteria. It is now easy to pass from this hierachy structure to a common finite state automata. We will now explain how to directly implement this structure by using the Bayesian temporal structures [8].

VI. Implementation

The Bayesian temporal structures are a way to implement by a probabilist method a finite state machine describing the general behaviour of a system. This was introduced in [8] to choose the behaviour of a robot among a set of behaviours. First of all, the general Bayesian program of the temporal structures is explained. Then, the link between the temporal structures and the hierachies of behaviours is done.

A. Bayesian temporal structures

The aim of the structure is to decide which behaviour to apply (choose the value of E_t) knowing the previous behaviour (E_{t-1}) and the current sensory-motor input $S_1 \ldots S_N$. We obtain the following Bayesian program:

Variables

$\mathbf{E_t}$: current state

$\mathbf{E_{t-1}}$: previous state

$\mathbf{S_1} \ldots \mathbf{S_N}$: relevante inputs.

Decomposition

$$P(\mathbf{E_t}, \mathbf{E_{t-1}}, \mathbf{S_1} \ldots \mathbf{S_N}) = P(\mathbf{E_t}) \times P(\mathbf{E_{t-1}} | \mathbf{E_t}) \times \prod_{i=1}^{N} P(\mathbf{S_i} | \mathbf{E_t}) \tag{11}$$

Parametrical forms

$P(\mathbf{E_t})$: we don't want to fix any *a priori* over the current state. We use then the uniform law.

$P(\mathbf{E_{t-1}} | \mathbf{E_t}), P(\mathbf{S_i} | \mathbf{E_t})$: this laws are fixed by histogramms. These tables can be fixed by the programmer, or learned by the robot. For our experiments, we have chosen the first solution.

Bayesian question

$$P(\mathbf{E_t} | \mathbf{E_{t-1}}, \mathbf{S_1} \ldots \mathbf{S_N}) \tag{12}$$

B. Direct implementation of the hierachies

Let $H = (\mathbf{B} = \{b_1 \ldots b_N\}, (\mathbf{R}_+, \mathbf{R}_-))$ be a hierarchy of N behaviours. We have to choose first which relevant inputs to use for the Bayesian temporal structures. Then, the histograms should be computed. The transition between two behaviours of the hierarchy are fllowed according to the credibility and relevance values. There is thus $2N$ sensor variables

$$\{\mathbf{S_1} \ldots \mathbf{S_{2N}}\} = \{\mathbf{C}(b_i), \mathbf{I}(b_i), \quad i \in \{1 \ldots N\}\} \tag{13}$$

$\mathbf{E_{t-1}} \setminus \mathbf{E_t}$	Lost	Corridor	Crosssing	Turn	Home
Lost	5	2	0	0	0
Corridor	1	5	2	0	0
Crosssing	1/2	1	5	2	2
Turn	1/2	1	0	5	0
Home	1	0	0	0	5

Fig. 5. Histogram $p(\mathbf{E_t} | \mathbf{E_{t-1}})$ unnormalized

$\mathbf{C} \setminus \mathbf{E_t}$	Lost	Corridor	Crosssing	Turn	Home
0	0	1	1	1	1
1	2	1	1	1	1

Fig. 6. Histogram $p(\mathbf{C}(\mathbf{lost}) | \mathbf{E_t})$ unnormalized

Then we should choose the values of the $3N$ histograms. The first histograms to be set are those giving the relation between the current and previous behaviours. The array chosen for our implementation is given on Fig. 5. We applied three criteria to choose the values.

- First of all, the favorite choise is to stay on the previous behaviour. We also set the values of the diagonal of the histogram to the higher value (we choose 5 for our experiments).

- Then, the links $\mathbf{R}_+$ and $\mathbf{R}_-$ are added. We favour the $\mathbf{R}_+$ relation by putting a higher value (we choose 2) than for $\mathbf{R}_-$ (we choose 1).

- We enable the links toward below behaviours. If there is a string of behaviours using relation $\mathbf{R}_-$ of a length l, we set the value between the two behaviours to $(\frac{1}{2})^{l-1}$.

- Finally, we forbid links between the other states by setting their values to 0.

The second histograms define the links between the credibility value expected when knowing the current state. A behaviour b is forbiden if its credibility value is FALSE. Thus, $p(\mathbf{C}(b) = FALSE | \mathbf{E_t} = b)$ is set to 0. On the opposite, this behaviour is favored when its credibility is TRUE. We thus set this probability to a high value (we choose 2). We have no *a priori* for the other cells, which are set to 1. An example of such an histogram is given fig. 6.

The last histograms are simpler. We only have *a priori* about the concerning behaviour. The probability $p(\mathbf{I}(b_0) | \mathbf{E_t} = b_0)$ is set equal to X, where X is the value of interest of the behaviour. The other cells are set to 1, since we don't have any *a priori* about them. An example is given Fig. 7.

VII. Experiments and results

We haven't given any details about the implementation of the behaviours. They are very simple. During our experiments, we have considered that they were perfectly correct. The experiments were executed to validate the general strategy, not the behaviours taken separately. Before the tests, the robot was driven away from its *home*, in a office away from the

$\mathbf{I} \setminus \mathbf{E_t}$	Lost	Corridor	Crosssing	Turn	Home
0	1	1	1	1	1
1	1	1	1	X	1

Fig. 7. Histogram $p(\mathbf{I}(\mathbf{turn}) | \mathbf{E_t})$ unnormalized

Fig. 8. The robot goes out of the office, and orientes itself in the way of the corridor

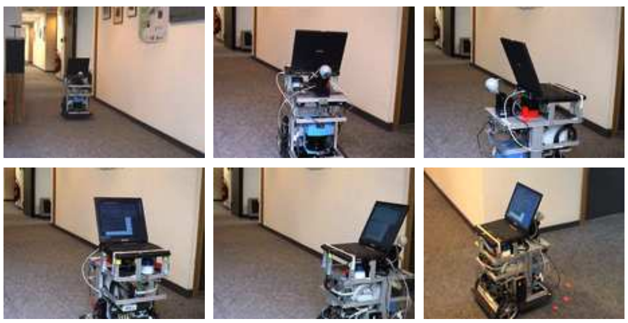

Fig. 9. The robot reach a crossing. It first rotate around itself to localize the red token, then chooses the left direction, and resume its corridor following in the left way

robotic hall where the robot should return by three corridors and four crossing. Red tokens were put at each crossing. The total length the robot must cover is approximately 150m. First of all, the robot go out off the office, using its lowest behaviours (fig. 8). The behaviour *corridor following* is then activated. When arriving at a crossing, the specific behaviour is chosen. The robot turns around itself to determine where is the maximum number of red tokens. On Fig 9, the maximum is detected on the left of the robot. The behaviour *rotate left* is then selected, the robot rotates, and resumes following the corridor. Finally, it reaches the last crossing (Fig. 10). During its self-rotatation, the robot detects more than ten red tokens. The last behaviour *homing* is activated. Using this very specific behaviour, the robot enters the home. The trip to the robot home ends.

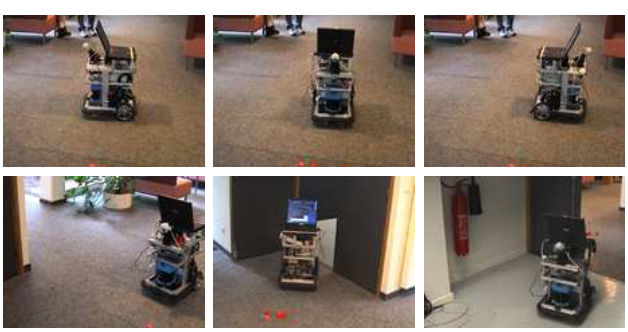

Fig. 10. The robot reach a last crossing. Ten red tokens are detected. The last behaviour is activated, and the robot enter the home.

VIII. CONCLUSION

In this article, a formal method to specify a structure for set of behaviours has been proposed. The hierarchy of behaviours provides an explicit set of criteria to choose which behaviour should be applied at each step. This structure has then been linked with the Bayesian temporal structures formalism, that makes possible to directly pass from a hierarchy of behaviours to the Bayesian program, and then to the robot implementation. We have proposed a simple example of a hierarchy using only low-level behaviours for navigating. The global strategy controls the robot so that it goes closer of the home following the corridor. When this simple behaviour is not precise enough, that is to say into a crossing, higher strategy level are used to determine which direction to choose. By using this structure, the robot is always able to use a valid behaviour. Moreover, it always use the lowest level of behaviour that allows it to reach its goal. The robot never uses high rate of computation or knowledge when lowest strategies are available.

The behaviours chosen where very simple. This experiments is only a first step in order to validate the proposed work. The robot reach its home *without never knowing where it is nor where the home is*. This is an interesting result. However the implementation is not useful by itself. The next step is to validate these results using higher level of navigation strategy, in particular using maps. We hope that we can join other mixed methods proposed in the state of art, by using this general and formal way of structuring behaviours.

A second direction of our next researches concerns the constructions of the automata that implement the hierarchy. To be as general as possible, we hope in the end for teaching the robot a high-level strategy by only provinding it an unstructured set of behaviour. A rigorous structure has been defined. It could now be learned automatically, by mimetic, or by random tries and rehenforcement.

REFERENCES

[1] P. Bessière, E. Dedieu, o. Lebeltel, E. Mazer, and K. Mekhnacha. Interprétation ou description (ii) : fondement mathématiques de l'approche f+d. *Intellectica revue pour l'association de la recherche cognitive*, 1:313–336, 1998.

[2] R.A Brooks. Intelligence without reason. *Proceedings of 12th International Joint Conference on Artificial Intelligence*, August 1991.

[3] R. Chatila and J.P. Laumond. Position referencing and consistent world modeling for mobile robot. *Proc. IEEE International Conference on robotics and automation*, pages 138–145, 1985.

[4] J.L. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. *AAAI workshop on spatial reasoning and multisensor fusion, Kaufmann Press*, 1987.

[5] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

[6] M.O. Franz and H.A. Mallot. Biometric robot navigation. *Robotics and Autonomous System*, 30:133–153, 2000.

[7] B.J. Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119:191–233, 2001.

[8] O. Lebeltel. Bayesian robots programming. *Autonomous robotics*, 2003.

[9] S. Thrun and A. Bücken. Learning maps for indoor mobile robot navigation. *Technical Report CMUCS-96-121*, 1996.

[10] N. Tomatis, I. Nourbakhsh, Arras K.O., and R. Siegwart. Simultaneous localization and map building: A global topological model with local metric maps. *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2001.