A Novel Action Selection Mechanism for Intelligent Service Robots

Il Hong Suh*, Woo Young Kwon*, and Sanghoon Lee**

{*The Graduate School of Information and Communications, **Department of Electrical Engineering and Computer Science}
Hanyang University, Seoul, Korea
(Tel : +82-2-2290-0392; E-mail: ihsuh@hanyang.ac.kr)

Abstract: For action selection as well as learning, simple associations between stimulus and response have been employed in most of literatures. But, for a successful task accomplishment, it is required that an animat can learn and express behavioral sequences.
In this paper, we propose a novel action-selection-mechanism to deal with sequential behaviors. For this, we define behavioral motivation as a primitive node for action selection, and then hierarchically construct a network with behavioral motivations. The vertical path of the network represents behavioral sequences. Here, such a tree for our proposed ASM can be newly generated and/or updated, whenever a new sequential behaviors is learned.
To show the validity of our proposed ASM, three 2-D grid world simulations will be illustrated.

Keywords: Action Selection, Decision Making, Behavior Selection, ASM

1. INTRODUCTION

An animat is an artificial organism – either a simulated animal or an animal-like robot – the structure and functionalities of which are based substantially on mechanisms observed in biological animals[1]. An animat must select an action that is appropriate to the situation in which the animat lives and learns how to survive. Thus, an animat with sensors and actuators is usually equipped with a action selection mechanism (ASM) that relates its perception to its actions and make it possible to adapt its environment[2].

In the field of animat research, one of the fundamental problems is to decide what to do next[3]. This problem is in the literature denoted as the action selection problem (ASP). In the ethological view, the ASP is the problem for an animal to design how to select its action so as to maximize its future expected genetic fitness[4]. But the ASP has proven to be a hard nut to crack due to (a) incomplete knowledge, (b) unpredictable environment and surrounding, (c) imperfect sensor and actuator, (d) limited resource[5].

The architecture of earlier systems, which were based on traditional AI planning methods, consisted of a sense-plan-act sequential cycle and the interaction between the sensing, planning, and action components. But traditional AI planning methods have some limitations, because They assume accurate knowledge of the world state provided by system sensors. This assumption is not valid due to a number of factors, such as changing world state, limited processing resources, and noisy, unreliable sensory information[5].

To overcome weakness of traditional AI approaches, a new reactive approach, called as “behavior-based AI”, has emerged. Brooks[6] has suggested a new architecture being called “subsumption architecture”, which are composed of competence modules with fixed priorities. This approach gives us an advantage, such as to fulfill a set of goals in a complex environment. To make an animat more life-like than subsumption approach, several researchers have proposed ethologically inspired model of action selection [5][7][8][9] [10]. Those models have showed good performances to imitate behavior of real life, since action selection in those models has been done based on competence modules with changing priorities. But most of those works generally involved ‘fixed’ pre-designed stimulus-response behavior systems and did not incorporate learning. Thus, they may not be appropriate in dynamic environments. Recently, several researchers has suggested ethologically inspired models of action selection that incorporate learning[11][12][13]. But much works remain to be done to cope with several shortcomings such as the lack of goal-handling ability.

In this paper, we suggest a novel architecture that allows learning to be combined with action selection, based on ideas from ethology. Furthermore, to overcome the lack of goal-handling capabilities, we improve current ethology-based architectures to deal with sequential behavior. Most of typical hierarchical structures organize actions in a hierarchy that range from high-level “nodes” or activities via mid-level composite action to detailed primitive nodes. Thus, only the primitive actions are actually executable. Our proposed ASM, however, can select the most appropriate motivation in given situation. And then, Our ASM can let a proper action be executed in each level within that motivation. As a result, Our ASM can choose correct sequential behaviors to satisfy a motivation and thus enables the system to learn necessary sequential behaviors.

2. ACTION SELECTION MECHANISM

ASMs can be generally classified as arbitration or command fusion architectures[15].

Arbitration mechanisms select one behavior, from a group of competence modules. Arbitration mechanisms for action selection can be divided into : fixed priority-based, winner-take-all, state-based. In fixed priority-based mechanisms, an action is selected based on a priori assigned

priorities[15].

The subsumption architecture proposed by Brooks[6] is typical fixed priority-based mechanism. This architecture consists of a series of behaviors, which constitute a network of hardwire finite state machines. Action selection consists of higher-level behavior overriding the output of lower-level behavior. Thus, each competence module of a level can be considered as having a priority, and high priority module suppresses low priority module. The control system is hard-wired directly in the structure of the behaviors and their interconnections, and can thus not be altered without redesigning the system. This type of architecture can be called as fixed priority-based arbitration architecture.

The other type of arbitration architecture is winner-take-all architecture, which is more flexible than subsumption architecture. Maes[9] and Blumberg[11] suggested this type of architecture. In this mechanism, action selection results from the interaction of a set of distributed behaviors that compete until one behavior wins others. Each competence module is considered to have priority varying under its own external and internal influences. Because these mechanisms are more flexible than fixed priority-based architecture, learning process can be easily incorporated.

Blumberg[11] suggested an architecture that allows learning to be combined with action selection, based on ideas from ethology. But, their work mainly focused on "do the right thing in a given situation". Therefore their structure only selects a single behavior to satisfy its need, and learns simple S-R associations. Note that behaviors to achieve a mission are consisted of a series of behaviors. Selecting a single behavior in a given situation is not enough to accomplish a mission.

Contrary to Bulmberg's model, our proposed ASM can decide both "what to do next" and "how that work can be achieved?". For this, a motivation that denotes a mission or goal competes with other motivations on the basis of internal needs and external stimuli. Next, a specific behavior to satisfy winner-motivation will be selected.

Our proposed ASM is a hierarchical organization of primitive modules named as Behavioral Motivation (BM) having their own stimuli(sensors) and behavior. To be more specific, we divide BMs into two types. Fist type is the Static BM(SBM) that denotes a motivation or mission. The connections among SBMs are fixed and cannot be changed until it is redesigned. Second type of BM is Dynamic BM(DBM), which can generate and learn sequential behaviors to satisfy the motivation.

Fig. 1 illustrates block-diagram of our ASM. In Fig. 1, Perception Filter(PF) system is a group of PF that filters external world information, and Internal State(IS) system is a group of internal influences such as drives. Fixed Action Pattern is a series of actions, and FAP system is a group of FAPs. Thus, BM has a link among PF, IS, and FAP. Finally, Learning system stores the information of past stimulus-action pairs, and it computes values of taking the action in the situation. Learning system enables the animat to learn new sequential behaviors, and to add these sequential behaviors to the BM system.

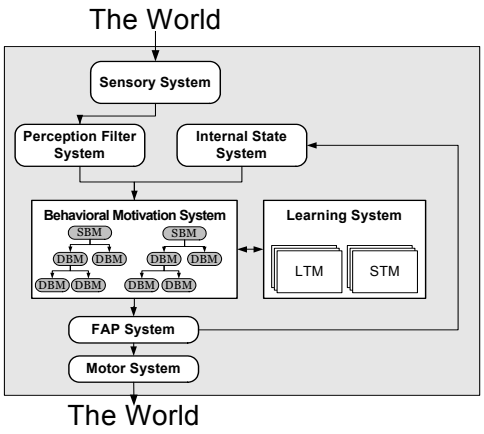

Fig. 1. Overall architecture

2.1 Static Behavioral Motivation(SBM)

An SBM implies a mission or a motivation, and values of SBMs will be used to determine what a motivation would be activated for given external stimuli and internal needs. For this, value of each SBM is computed by combining the value of IV with that of PF, and then by inhibiting other values of SBMs. In addition, an SBM receives a feedback effect from the DBM group under this SBM. The value of an SBM is calculated by using the equation given by

$$V_{SBM\,i(t+1)} = \sum V_{IV_{jt}} + \sum V_{PF_{kt}} - \sum_{all\ \mathrm{SBM}} \left(V_{SBM_{lt}} I_{il} \right) + effect^{i}{}_{DBM}, \qquad (1)$$

i : index of where the i th SBM

j : index of related IS

k : index of related PF

I_{il} : inhibitory Gain that $SBM\ i$ applies against $SBM\ l$

l : index of same level SBM(inhibition)

$effect^{i}{}_{DBM}$: feedback value from DBM under the i th SBM.

Here, $effect^{i}{}_{DBM}$ is given as

$effect^{i}{}_{DBM} = w_i V_{DBM_m}$,

m : index of maximum valued DBM under i th SBM.

w_i : weight of feedback value from DBM under the i th SBM.

The term $effect^{i}{}_{\mathrm{DBM}}$ means the strength indicating how the goal can be easily achieved for a given current state of the environment. Thus, the value of an SBM may be high not only when needs of the SBM become more important than those of other SBMs, but also when its goal is believed to be easily achieved for the given current state of the environment. Each SBM has a group of DBMs implying sequential behaviors to satisfy the SBM(or motivation). SBMs are organized into a pre-designed flat-network as in Fig. 2.

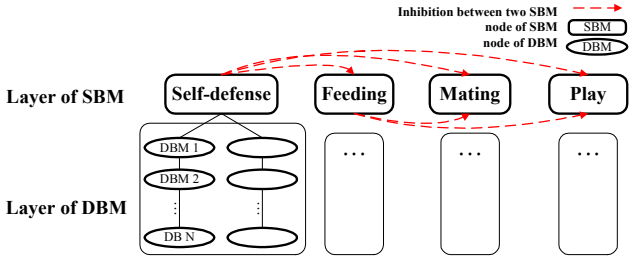

Fig. 2. An Examplar configuration of SBM and DBM

2.2 Dynamic Behavioral Motivation(DBM)

To accomplish a goal(or mission), an animat must generate a series of behaviors and select the most appreciate one. For this, DBMs are organized into flexible hierarchical network that can be changed by learning process.

A DBM has its own activation-value that depends on the values from PF, parent node, and releasers. A DBM outputs its value to child node, while relevant stimulus is incoming. The schematic of a DBM is illustrated in Fig.3. The activation-value is accumulated through the path, while relevant stimulus presents. Releasers play a role of blocking the flow of activation-value. The value of a DBM is given as

$$V_{DBM\,i} = \left(V_{DBM_{i-1}} + V_{PF_j}\right) STEP\left(\sum_{k=1}^{m} V_{Releaser_k}\right) \tag{2}$$

i : index of this Node

j : index of related PF

k : index of related releaser

$$STEP(x) = \begin{cases} 1, & \text{for } x>0 \\ 0, & \text{for } x=0 \end{cases}$$

The appropriate DBM will be selected by choosing maximum-valued DBM in a DBM group. Following is the equation to select a DBM;

$$selectedDBM = \underset{i \in all \text{ DBM under SBM}}{\arg\max} \left(DBM_i\right) \tag{3}$$

The path from the top level DBM to the bottom level DBM consists of sequential behaviors. By performing these sequential behaviors, the motivation (or SBM) can be satisfied.

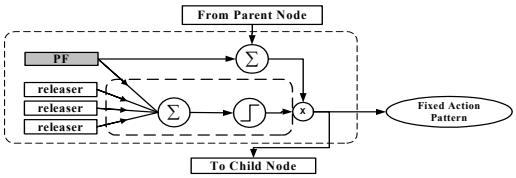

Fig. 3. A Primitive node of a DBM

2.3 Action Selection Process

Our ASM selects the most appropriate action(or FAP) based on its internal need and external environment on every cycle. To do this, the BM system selects relevant SBM and DBM. The SBM means what a mission must be achieved, and the DBM means what an action must be selected to satisfy a given motivation. The value of a SBM(or mission) depends not only on its internal needs, but also on how easily the goal will be archived. Thus, the value of a SBM reflects their internal need and feedback effect from its DBM. The selecting process for SBM and DBM is summarized as follows;

- A maximum-valued DBM is selected in each DBM group.
- Value of each SBM is computed by using Eq. (1), and then is compared with those of other SBMs to select the maximum-valued SBM.

After a maximum-valued SBM is chosen, one of the following two processes will be activated to select an action; 'exploit' and 'explore'. An exploit-process is performed when a BM system has enough knowledge to satisfy its motivation. The exploit-process is performed by executing the most appropriate FAP(or action) for a given situation.

Otherwise, an explore-process is performed (i) when a BM system has no knowledge to satisfy its motivation, or(ii) when a BM system has a little knowledge to satisfy its motivation. Especially, situations with no prior knowledge are divided as follows;

- When the selected SBM has an empty DBM group
- When the selected SBM does not include a DBM that could be matched with the current situation.
- When a DBM is selected several times without reaching a goal.

Like an exploit-process, an explore-process should decide an action. Specifically, a PF is randomly selected among PFs that have non-zero values. An FAP is also randomly selected among all FAPs. Then, the selected FAP is executed and the selected PF and FAP will be stored in STM. After several cycles, if an animat satisfies the given motivation, past relations between PF and FAP that have been stored in STM will be transferred to LTM as will be described in 3.1.

Now, when a little prior knowledge is available for an SBM, such a knowledge, which will be a group of relations of PFs having non-zero values and all FAPs, will be involved to select an action. That is, one relation of PF and FAP among a group of known relations will be selected and the corresponding FAP will be chosen based on the probability given as the reliability value of LTM. Fig. 4 illustrates overall process of our action selection. Here, it is remarked that to learn different strategies for different motivations, each SBM has its own LTM and STM.

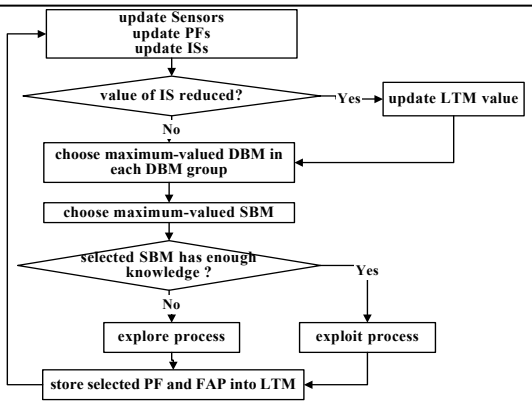

Fig. 4. The Process of Action Selection

2.4 Input Variables for SBM and DBM

■ **Perception Filter(PF)**

A Perception Filter (PF) identifies significant stimuli or events from input sensors, and output a value, which represents its strength and relevance. In other words, each PF outputs a continuous value which typically depends on existence of specific stimuli, and some quantitative measure such as distance. SBM and DBM can be made more or less sensitive to the presence of the relevant stimulus.

■ **Releaser**

In our proposed ASM, the DBM will convey its value to the child node, while relevant stimulus is incoming thru corresponding PF. But, the flow of activation-value will be blocked, if relevant stimulus disappears. That is, relevant stimulus plays a role of releasing activation-value. This PF, which releases and/or blocks the value-flow, is called as 'releaser'. By releasers, our proposed ASM can deal with sequential behaviors. Fig. 5 shows sensory bottleneck[5]. In order to activate the node 5 stimulus E is required. But the node 1 and the node 3 does not pass stimulus E. Therefore, the node 5 may not be activated. To avoid such a blocking of stimuli, every nodes must pass the stimulus to their lower nodes, which is valid in Our ASM by use of releasers.

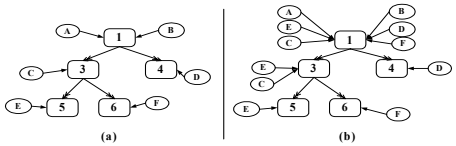

Fig. 5. An example of sensory bottleneck

■ **Internal State**

An Internal State (IS) is used for modeling drives such as hunger or thirst. The FAP, which reduces a certain IS or satisfies drives, is called as consummatory action and other FAPs are called as appetitive actions. The value of IS is changed after an FAP is executed. The relation of FAP and IS is defined by the gain. Based on Hull's theory[16], the reduction of drive can lead to a learning process. Therefore, the reduction in the value of IS by executing a consummatory action is used as a reinforcement signal for the learning. The value update equation of IS is given by

$$IS_i = IS_i + gain_{ij} \qquad (4).$$

i : index of IS

j : index of selected FAP

3. INTEGRATION OF LEARNING INTO ASM

3.1 Learning Process

The Learning system consists of short-term memory (STM) and long-term memory (LTM). In the STM, stimulus-action pairs are recorded along the time. All stimulus-action pairs in the STM are assumed to be appetitive behaviors to get rewards. With references to the time at which a reward is received, stimulus-action pairs near the reference time should be more credited than stimulus-action far from the reference time. This can be reflected as a reliability value, V, given by

$$V(s_i,a_i,s_{i+1}) \leftarrow V(s_i,a_i,s_{i+1}) + \frac{\eta(1-V(s_i,a_i,s_{i+1}))}{(N-i+1)^{\lambda}}, \qquad (5)$$

where s_i is the index of the i^{th} stimulus, a_i is the index of behavior, η is a learning rate, λ is a decay rate, N is the size of a STM, and $(N-i)^{\lambda}$ is weightings of distance from reference time. Fig. 6 shows pseudo-code of STM and LTM operation for exploration of learning process.

```
N = SizeSTM
repeat for each episode
{
   initialize ( STM )
   repeat for each step of episode
   {   choose action and state a,s using Boltzman exploration
       store a, s to STM
       take action a
   }
   loop i = 1 to N
          V(s_i,a_i,s_{i+1}) ← V(s_i,a_i,s_{i+1}) + η·(1 - V(s_i,a_{i+1},s_{i+1})) / (N - i + 1)^λ
}
```

Fig. 6. The Pseudo-code of learning process.

3.2 Integration of learning

After the animat performs some sequential behaviors and receives a reward, stimulus-action pairs that consist of sequential behaviors will be stored in the LTM. Thus, the LTM may include several paths to accomplish task. LTM entries with values, which exceeding a certain threshold will be added to BM system. Fig. 8 shows an example of interaction between the Learning system and the BM system. Note that Fig. 7-a shows the LTM state after some trials were performed, and the value of 'S_G-B_G-goal' exceeds a threshold.

Because the LTM entry 'S_G-B_G-goal' has not been included in corresponding hierarchical structure, this entry will be added to the branch of the SBM.

After more trials are performed, the LTM may be changed as in Fig. 7-b. In Fig. 7-b, there are two LTM entries with values to exceed the threshold. Because the entry ('S_G-B_G-goal') has been already included in the hierarchical structure, another entry ('S_3-B_3-S_G') will be added. To reach the goal, the action B_3 in 'S_3-B_3-S_G' must precedes the action in 'S_G-B_G-goal'. Thus, the position of that entry will be the parent node 'S_3-B_3-S_G'. After many trials, a new appetitive behavior to reach goal can be added in the structure as shown in Fig. 7-c.

4. EXPERIMENTS

To show the validities of our proposed ASM, several experiments are performed for a 2-D grid world. The animat can move to directions such as "north, south, east, west" and can get information within the sensory range of 4x4 grid world. Features in experiment are given as relative locations of the animat, blocks and bugs. Here, when the animat push block, the block will move to the same direction, of the animat movement.

The rules for the animat to learn are given as;

Rule 1 : When the animat, block and bug are in a straight line, the animat can eat the bug, by pushing the block the, and then receives rewards. (Fig. 8-a)

Rule 2 : When the animat, block and bug are not in a straight line , the animat moves to the position at which rule 2 can be fired. (Fig. 8-b)

Rule 3 : When the animat not contacted with the block, animat moves to the position at which rule 2 is fired. (Fig. 8-c)

To learn the Rule 1, 50 trials were performed in the 3x3 grid world. To learn the Rule 2, same grid world for the Rule 1 was used, and 100 trials were performed. Finally, the Rule 3 was learned in 4x4 grid world by 1000 trials. From Fig. 9 and 10, it is observed that DBMs of the hierarchical structure for a desired mission(or sequential behaviors) have been successfully formed, after the Rule 1, Rule 2, and Rule 3 have been learned respectively.

5. CONCLUSION

Selecting and learning appropriate actions to survive in its environment are the most important abilities in an animat. For this, we proposed a hierarchical organization of competence modules called as SBM and DBM. The SBM was used to select the most appropriate motivation in a given situation. And, the DBM was used to select a behavior that could satisfy its motivation. By use of releasers to block the activation-values of DBMs, a hierarchical group of DBMs can generate sequential behaviors. Thus, our proposed ASM can not only select the most appropriate behavior in a given situation, but also deal with sequential behaviors. Furthermore, our proposed flexible hierarchical network can add learned behaviors.

To show the validity of the proposed ASM, experiments in 2-D grid world were performed. In these experiments, three simple rules were successfully learned and established in the hierarchical structure.

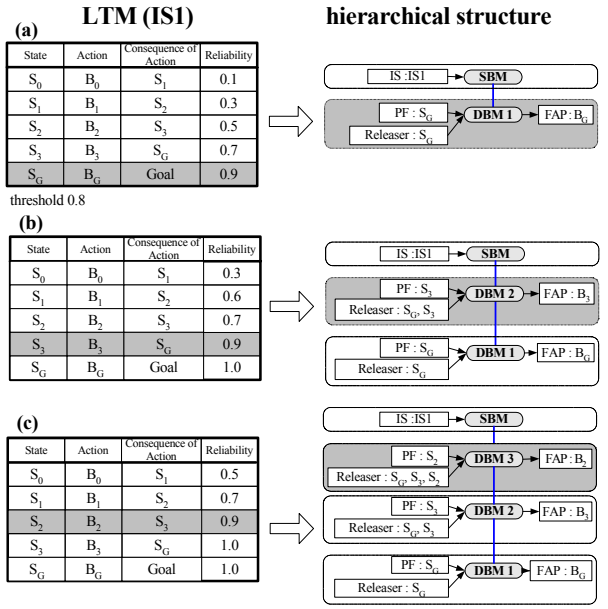

Fig. 7. An example of interaction between LTM and hierarchical structure.

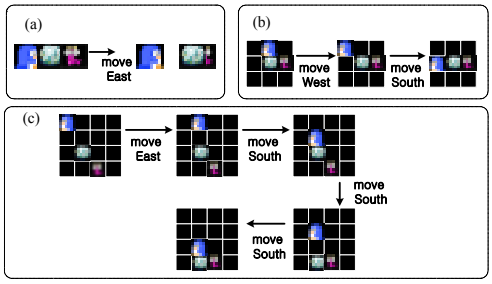

Fig. 8. The rules to learn

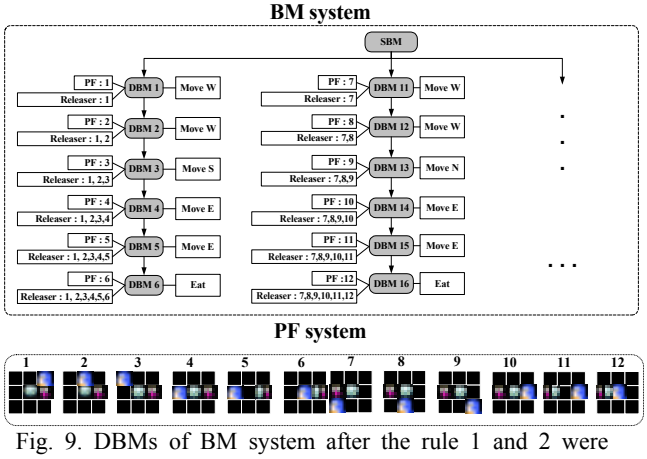

Fig. 9. DBMs of BM system after the rule 1 and 2 were learned

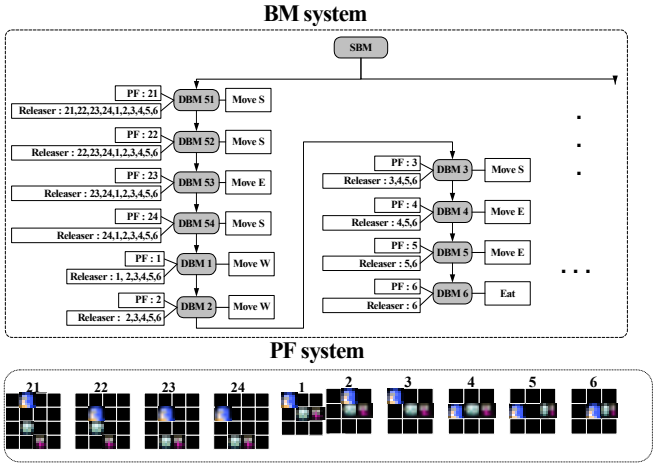

Fig. 10. DBMs of BM system after the rule 3 was learned

REFERENCES

[1] A. Guillot and J.A. Meyer, "Computer simulations of adaptive behavior in animats.", Proceedings Computer Animation'94., IEEE Computer Society. ,1994.

[2] S. W. Wilson, "The Animat Path to AI.", In From Animals to Animats, First International Conference on Simulation of Adaptive Behaviour, MIT Press, Cambridge, 1991.

[3] P. Maes, "Modeling Adaptive Autonomous Agents.", Artificial Life Journal, Vol. 1, No. 1 & 2, pp. 135-162, MIT Press, 1994.

[4] P. Pirjanian, "An Overview of System Architectures for Action Selection in Mobile Robotics.", Tech-report, Laboratory of Image Analysis, Aalborg University 1997.

[5] T. Tyrrell, "Computational Mechanisms for Action Selection." , Ph.D. Thesis, Centre for Congitive Science, University of Edinburg ,1993.

[6] R. Brooks, "A Robust Layered Control System For a Mobile Robot", In IEEE Journal of Robotics and Automation, pages 14 23, April, 1986.

[7] J.K. Rosenblatt and D.W. Payton, "A Fine-Grained Alternative to the Subsumption Architecture for Mobile Robot Control.", Proceedings of the IEEE/INNS International Joint Conference on Neural Networks, Vol. II, pp. 317-323, 1989.

[8] E. Spier and D. McFarland, "A Finer-Grained Motivational Model of Behaviour Sequencing.", In From Animals to Animats 4 : Proceedings of SAB96, 1996.

[9] P. Maes, "A Bottom-Up Mechanism for Behavior Selection in an Artificial Creature.", In From Animals to Animats, First International Conference on Simulation of Adaptive Behaviour, MIT Press, Cambridge, 1991.

[10] B.M. Blumberg, "No Bad Dogs : Ethological Lessons for Learing in Hamsterdam", and Interactive Createures, MIT ,1997.

[11] B. M. Blumberg, "Old Tricks, New Dogs.", Ethology and Interactive Createures, MIT ,1997.

[12] C. M. Witkowski, "Schemes for Learning and Behaviour: A New Expectancy Model.", Ph.D. Thesis, University of London, 1997.

[13] M. Humphrys, "Action Selection methods using Reinforcement Learning.", PhD Thesis, University of Cambridge, England, 1996.

[14] D.C. Mackenzie, R.C. Arkin and J.M. Cameron, "Specification and Execution of Multiagent Missions", Autonomous Robots, 4(1), 1997.

[15] P. Pirjanian, "Behavior Coordination Mechanisms – State-of-the-art", Tech-report IRIS-99-375, Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California, October, 1999.

[16] C.L. Hull, Principle of behavior, new York : Appleton-Century-Crofits, 1943.