Bull. Nov. Comp. Center, Comp. Science, 24 (2006), 57–72
© 2006 NCC Publisher

Animat control system based on semantic probabilistic inference

A. V. Demin, E. E. Vityaev

Abstract. The paper presents a model of adaptive behavior of an autonomous adaptive agent (an artificial organism) based on the semantic probabilistic inference and the functional system theory by P. K. Anokhin. The main distinction of this model is the possibility for automatic generation of new subgoals, which allows us to solve more complex multi-level tasks. An autonomous adaptive agent has been created on the basis of this model, and a number of experiments have been carried out in order to train it and to compare it with the existing approaches based on neural networks and reinforcement learning. The results of comparison have shown that the proposed model learns and acts more efficiently.

1. Introduction

Recently, active research is carried out in the field of adaptive behavior that involves studies of the fundamental principles enabling natural or artificial organisms to adapt to variable environment. One of the main approaches in this research direction is to create and study agents (computer programs or robots) whose behavior is based on the principles of a living organism behavior. These agents are called "animats" (animal + automat = animat).

This work offers a general scheme of an adaptive animat control system which includes an architecture based on a hierarchy of functional systems and subgoals and a training algorithm which uses the semantic probabilistic inference and the possibility to generate new subgoals automatically. On the basis of the proposed model, an elementary animat and its environment have been implemented in the form of a computer program. Using this program, we have carried out a number of experiments in the animat learning and made test comparison with the existing approaches based on neural networks and reinforcement learning.

2. The theory of functional systems

The structure of the control system we propose is based on the theory of functional systems developed in 1930–70s by the famous soviet neurophysiologist P.K. Anokhin [9–10]. According to this theory, a functional system that achieves some results beneficial for an organism (for example, satisfaction of its needs) is considered to be a unit of this organism's activity. A

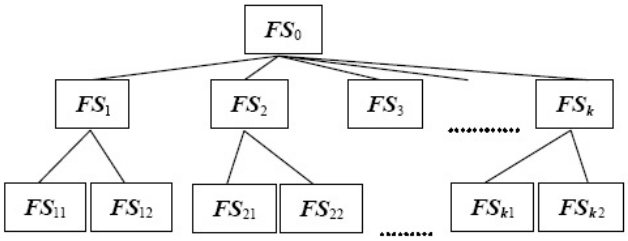

Figure 1. Structure of the control system

functional system with goal-seeking behavior is organized according to two rules: the sequence and hierarchy of the results. The sequence of the results is built on the principle of a "dominant": the dominating need stimulates the dominant functional system and creates the behavioral act aimed to satisfy it. With respect to the dominating functional system, all other functional systems are structured by the principle of "the hierarchy of results": the result of one functional system constitutes a part of the result of another system.

The central mechanisms of a functional system responsible for its goal-seeking behavior have the same structure. The initial stage of a behavioral act of any complexity is *afferent synthesis* that includes synthesis of motivation, memory and information about the environment. As a result of afferent synthesis, all possible ways of achieving the goal in this situation are evoked from the memory. At the stage of *decision-making,* only one particular way of action is selected according to the initial need. To provide for achievement of the results, an *actions results acceptor* is being created beforehand which is a model of the parameters of the expected result. Each action is simultaneously followed with signals about achievement of the result, called *backward afferentation.* Actions on the goal achievement are taking place until the parameters of the action result perceived by the central nervous system in the form of the corresponding backward afferentation would perfectly comply with the properties of the actions results acceptor.

The theory of systemogenesis that studies the mechanism of the functional system formation, is a special branch of the general theory of functional systems. In this paper, we also study the mechanism of formation of a new functional system on the basis of the subgoal identification.

3. Animat control system

In accordance with the theory of functional systems, we will assume that the animate control system being modeled has a hierarchical structure, where separate functional systems constitute the basic elements of the control system. In this structure, the upper level functional systems set the goals for the lower level systems. At the same time, we can think that each functional system solves the task of achieving the goal using the same methods as other functional systems. Figure 1 shows the structure of the animat control system.

The tasks of a functional system are as follows:

- Given a goal (subgoal) and available information about the environment and the state of the functional system, find an optimal way of achieving this goal.
- If, on the basis of a forecast, an action is found that guarantees achievement of the goal, then give an instruction to perform this action.
- Control the correctness of performing the action, that is check the compliance between the achieved and the desired results.

4. A model of a functional system operation

Figure 2 shows the model of a functional system operation based on the works [1–2]. Let us think that a goal P is set to the functional system at some point of time (this is done in the form of a request to the functional system to achieve the goal P). Information about the environment is also supplied in the form of the situation description $P_1, ..., P_m$. In the process of afferent synthesis, all information related to achievement of the goal P (it is stored as a set of patterns of the form $\langle P_1, ..., P_k, A\rangle \to P$) is evoked from memory. Note that only those patterns are evoked, in which the situation properties are met, i.e., all properties of a situation P_i that are included in the condition $P_1, ..., P_k$ of some pattern should also be included in the situation description $P_1, ..., P_m$.

The pattern conditions $P_1, ..., P_k$ include not only the properties of the situation, but also subresults $P_1, ..., P_n$ that should be achieved in order to achieve our goal P. To achieve the subgoals, we send the corresponding requests down the hierarchy (in Figure 2 it is indicated by the double arrow pointing down). These requests activate all information related to achievement of these subgoals in the same situation (as above) in lower level functional systems, which, in turn, may require achievement of other goals at even lower levels, and so on. If some subgoal cannot be achieved in this situation (there are no patterns predicting its achievement in this situation), then denial is received in reply to the request and the corresponding pattern is excluded from consideration.

In the block of afferent synthesis $\langle P_1, ..., P_k, A_i\rangle \to P$, the pattern activation procedure evokes from memory a set of actions A_i (including the actions required to achieve the subgoals) that may lead to achievement of

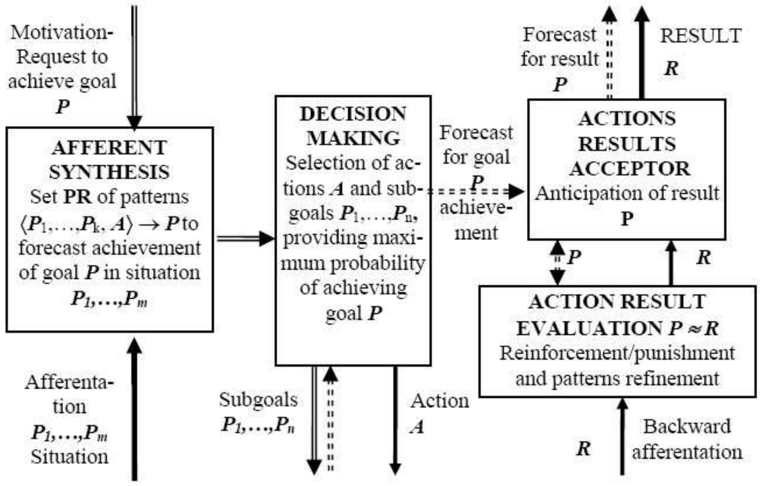

Figure 2. The model of the functional system operation

the goal P. This set of actions, together with evaluations of conditional probabilities for achievement of the goal and subgoals, is passed to the decision-making block. Here all actions A_i are examined (together with the patterns $\langle P_1, ..., P_k, A_i\rangle \to P$ activating them and with the hierarchy of subgoals and corresponding actions) and an action is selected so that it provides the maximum estimate for the probability to achieve the goal P, taking into account the probabilities to achieve the subgoals. After that the action A and all actions necessary to achieve the subgoals are launched. At the initial stage of learning, when there are no rules yet, or there are no rules applicable in this situation, the action of the corresponding functional system is randomly selected and no forecast exists.

The forecast for the anticipated result P and all subresults for all subgoals is sent to the actions results acceptor. Besides, the forecast for subresults in all lower level functional subsystems is also sent to the corresponding actions results acceptors.

The data on the achieved result R are sent to the result evaluation block of the actions results acceptor for the forecasted and obtained results to be compared. If the forecast coincides with the results with the given degree of accuracy, then the pattern selected in the decision-making block is reinforced, otherwise it is punished. Reinforcement/punishment consists in increase/decrease of the pattern's conditional probability. Besides, the set of rules is refined more precisely after each action, as was shown in [1–2]. If after this refinement a pattern is found having higher conditional probability than the earlier used one, then the new pattern will be used for forecasts and decision-making.

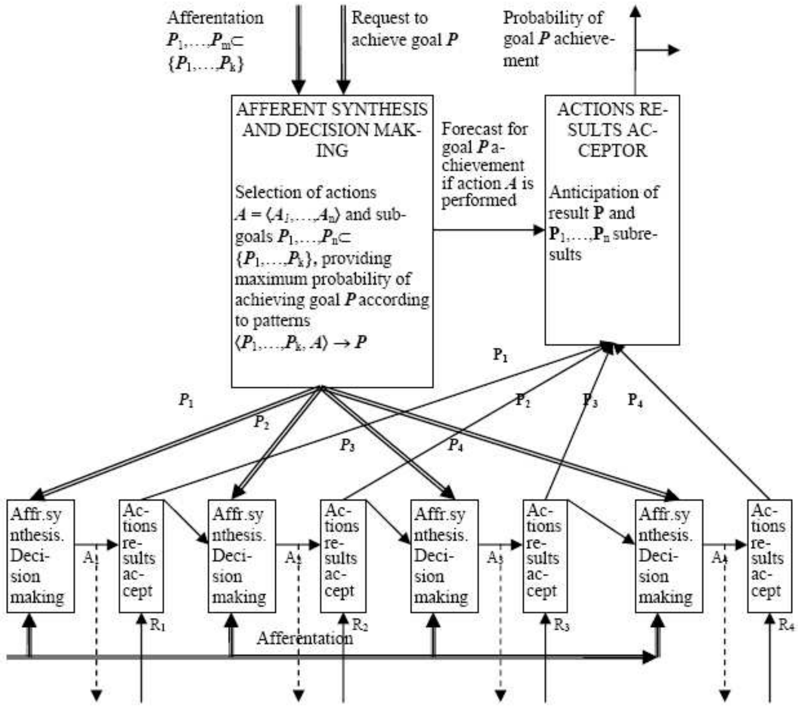

Figure 3. Two levels of the functional systems hierarchy

The semantic probabilistic inference [6] allows us to find a set of patterns PR in the form of $\langle P_1 \ \& \ ... \ \& \ P_k \ \& \ A\rangle \rightarrow P$ that predict with maximum conditional probability the result P of the action A under the conditions $\langle P_1, ..., P_k\rangle$.

5. The hierarchy of functional systems

Let us present a more schematic diagram of a functional system in Figure 3, where, instead of the whole hierarchy of functional systems, we show only two its levels. A functional system is not a construction determined once and for all. They change their form according to goals. Goals and subgoals in their turn also depend on successful achievement of the final goals. Let us show how goals and subgoals may be created automatically with the help of patterns.

Let us extend the notion of a result so that it can be generated automatically while working in a complex probabilistic environment:

a) The result should have the property of branching: if some result is obtained, then further actions may be determined ambiguously;

b) The result should contain a set of attributes which show that the goal of this sequence of actions is achieved and it is possible to move to the next sequence of actions, i.e. the result is a fixation of action completeness, ensuring the possibility of performing a certain subsequent action.

The condition a) of the above definition is naturally fixed by patterns, because they can make a good forecast for the results of a sequence of elementary actions (of a given level), if this sequence is a standard one, which means that, once started, it goes on without changes until a certain result is obtained. In this case the patterns can forecast realization of this sequence of actions with a high probability. Figure 3 shows them as actions A_1, A_2, A_3, A_4 leading to the results R_1, R_2, R_3, R_4. The actions results acceptor matches R_1, R_2, R_3, R_4 with those predicted by patterns and, if they coincide, gives the answers P_1, P_2, P_3, P_4 to the requests P_1, P_2, P_3, P_4. The answers about achieving the goals are passed to the inputs of other units (the arrows from the actions results acceptors pointing down to the subsequent units). These answers will be automatically included into the pattern conditions for subsequent actions, because a signal telling that the preceding action is completed increases the probability for the subsequent action to be completed. According to the definition of the semantic probabilistic inference [6], any signal increasing the probability of a forecast is automatically included into the pattern conditions.

The condition b) is also met, because the signals from backward afferentation proving that the preceding action has been really completed increase the probability of achieving the results of the subsequent action. The process of automatic generation of goals and subgoals at a neural level is described in more detail in [3].

6. Model description

1. Model description. Let us show in Figure 2 how the animat is working (according to the scheme of a functional system [4–5]). We assume that the animat control system works in a discrete time $t = 0, 1, 2, \ldots$. Let our animat have a set of sensors $S_1, S_2, ..., S_n$ characterizing the conditions of its internal and external environment, and a set of possible actions $A_1, A_2, ..., A_m$. From its set of sensors, we separate the sensor SA which provides us with information about the performed action. We suppose that the history of animat's actions is stored in a data table $X = \{X_1, ..., X_t\}$, where the t-th line contains the values $S_1^t, S_2^t, ..., S_n^t$ of the sensors $S_1, S_2, ..., S_n$ at the moment t: $X_t = \{S_1^t, S_2^t, ..., S_n^t, SA_t\}$. On the set X, we define a set of predicates $P_0 = \{P_1(t), ..., P_k(t), PA_1(t), ..., PA_m(t)\}$, where $P_i(t)$ are *sensor predicates* which specify some conditions on sensors' data at the moment

t and $PA_i(t) \Leftrightarrow (SA(t) = A_i)$ are *activating predicates* which show that the action A_i was performed at the moment t.

Let us introduce the concept of a *predicate-goal* $PG(t) = P_1(t)\&P_2(t)\&...\&P_l(t)$ which realizes the condition for achievement of the goal at the moment t.

Each functional system FS has a corresponding goal G it should achieve and a predicate-goal PG which characterizes the condition for achievement of this goal.

Each functional system FS has its own set of predicates $P = P_0 \cup \{PG_1, ..., PG_n\}$, where PG_i are predicates-goals corresponding to the goals of its subordinate functional systems of the lower level. Each functional system FS contains a set of patterns PR of the form $P_1, ..., P_k, PG_1, ..., PG_n, PA \rightarrow PG$. Each of these patterns is characterized by an estimate p for the probability of achievement of the goal PG in case the pattern condition is met.

Let us assume that at the moment t the functional system FS receives a request to achieve the goal PG. Then all patterns whose conditions are met at the moment t are extracted from set of patterns PR. If a pattern condition contains the predicates-subgoals $PG_1, ..., PG_n$, then the functional system sends a request to achieve these subgoals down the hierarchy. From the set of extracted patterns, we choose that which provides the maximum estimate f for the probability to achieve the goal. The estimated probability f of the pattern $P_1, ..., P_k, PG_1, ..., PG_n, PA \rightarrow PG$ is calculated as follows:

$$f(PG|P_1, ..., P_k, PG_1, ..., PG_n, PA) = p \cdot f(PG_1) \cdot ... \cdot f(PG_n),$$

where p is the estimated probability of this pattern, $f(PG_i)$ are the estimated probabilities to achieve the subgoals.

If all conditions of the selected pattern are met, then the action A is launched. If the set of patterns PR is empty or there is no pattern applicable in this situation, then the action is randomly chosen from the whole range of actions available.

After the action is performed, the sensor data are renewed, the action result is estimated and the set of rules PR is refined (see below).

2. Evaluation of the action results. Each functional system FS stores the estimates for the results of its actions $d(t)$ for each moment t. Let us describe the method of derivation of these estimates.

We assume that at the moment t_0 the functional system FS received a task to reach the goal G, and the result R was obtained after the goal was achieved at the moment t_1. Then the estimates for the results of actions $d(t)$, starting from t_0 up to t_1, are calculated as follows:

$$d(t) = r\frac{t - t_0}{(t_1 - 1) - t_0}, t_0 < t < t_1,$$

where r is a function of estimating the quality of the obtained result,

$$r = \begin{cases} 0, & \text{if } PG = 0 \\ mes(G, R), & \text{if } PG = 1 \end{cases},$$

where $mes(G, R)$ is a degree of proximity between the obtained result R and the goal G. We use

$$mes(G, R) = \begin{cases} 0, & \text{if } G \neq R \\ 1, & \text{if } G = R \end{cases}$$

in our experiments.

3. Rule generation. To obtain the set of patterns PR used by functional system FS, we use the semantic probabilistic inference [6].

The semantic probabilistic inference allows us to find all patterns of the form $P_1, ..., P_n \to P_0$ which predict the predicateP_0 with maximum probability. Inference is made on a set of training data Y with the use of the given set of predicates $\{P_1, ..., P_m\}$.

This method is based on the following definition of a probabilistic pattern proposed in [7].

A rule $P_1, ..., P_n \to P_0$ is a *pattern*, if it meets the following conditions:

1. $p(P_1, ..., P_n) > 0$,
2. $\forall \{P_1, ..., P_k\} \subset \{P_1, ..., P_n\} \quad p(P_0|P_1, ..., P_n) > p(P_0|P_1, ..., P_k)$.

Here p is the estimate for the conditional probability of the rule.

Let us introduce the concept of a rule *refinement*. A rule $P_1, ..., P_n, P_{n+1} \to P_0$ is a *refinement* of a *rule* $P_1, ..., P_n \to P_0$, if it is obtained by addition of an arbitrary predicate P_{n+1} to the rule's premise $P_1, ..., P_n \to P_0$, and $p(P_0|P_1, ..., P_{n+1}) > p(P_0|P_1, ..., P_n)$.

The algorithm of the semantic probabilistic inference is as follows:

- At the first step, a set of refinements of the rule $\to P_0$ is generated (that is of the rule with empty premise). This set will consist of rules of a unit length of the form $P_1 \to P_0$, for which $p(P_0|P_1) > p(P_0)$.
- At the k-th ($k > 1$) step, a set of refinements is generated for all rules created at the previous step. That is for each rule $P_1, ..., P_{k-1} \to P_0$, generated at the $(k-1)$-th step, a set of rules of the form $P_1, ..., P_{k-1}, P_k \to P_0$ is created so that $p(P_0|P_1, ..., P_{k-1}, P_k) > p(P_0|P_1, ..., P_{k-1})$.
- It is checked whether the obtained rules are patterns. The rules that do not comply with the pattern conditions are eliminated.
- The algorithm stops when it is impossible to refine any rule.

In order to avoid generating statistically insignificant rules, an additional criterion is introduced — estimate of statistic significance. Rules that do not

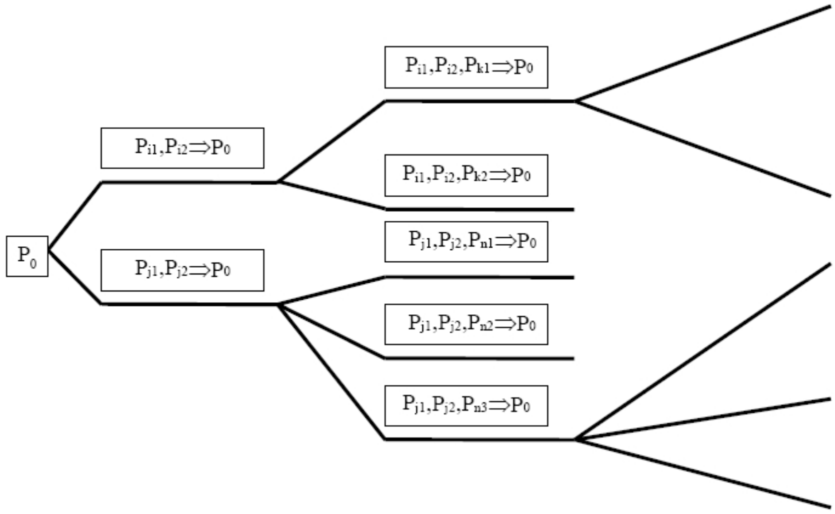

Figure 4. The tree of semantic probabilistic inference

comply with this criterion are eliminated even if they have high accuracy on the training set. To evaluate statistical significance, Fisher's criterion (exact Fisher's criterion for conjugation table) is used in our algorithm.

It is evident that all rules obtained with the help of this algorithm are patterns. Figure 4 shows the inference tree which describes this process.

To find all patterns $P_1, ..., P_k, PG_1, ..., PG_n, PA \to PG$ predicting achievement of G with maximum probability, a tree of the semantic probabilistic inference is built over the set of data on the history of actions of the animat X and the set of action estimates $d(t)$ with the use of the set of predicates P from this functional system. The estimate of conditional probability p for a rule is calculated as follows: $p = \sum\limits_{i \in I} d_i \Big/ ||I||$, where I is a set of moments when this rule can be applied.

4. Subgoal identification. Initially the animat control system has a hierarchy of its functional systems given a priori. In the simplest case it may consist of only one functional system. In the process of functioning, the control system may automatically identify new subgoals and generate the corresponding functional systems.

Let us define a subgoal as a situation which significantly increases the probability of achievement of a higher-level goal, and actions following this situation cannot be defined unambiguously.

In order to identify a subgoal, the set of rules PR of the functional system is analyzed. We denote by $Cond(R) = \{P_1, ..., P_n\}$ the set of sensor predicates in the premise of the rule $R = P_1, ..., P_n, PA \to PG$. The situation described by the predicate $PG_{new} = P_1 \ \& \ ... \ \& \ P_k$ is a subgoal G_{new}, if the following conditions are satisfied:

1) $\forall R_1 \in PR$ such that $\{P_1, ..., P_k\} \subseteq Cond(R_1)$ and $\forall R_2 \in PR$ such that $Cond(R_2) \subset Cond(R_1)$ and $\{P_1, ..., P_k\} \not\subset Cond(R_2)$, we have $p(R_1) - p(R_2) > \delta$.
2) $\exists R_1 = P_1, ..., P_n, PA_1 \to PG, \quad R_1 \in PR$, and $R_2 = P_1, ..., P_m, PA_2 \to PG$, $R_2 \in PR$, such that $\{P_1, ..., P_k\} \subseteq Cond(R_1)$, $\{P_1, ..., P_k\} \subseteq Cond(R_2)$ and $A_1 \neq A_2$.

The first condition says that addition of this situation to the conditional part of the rules should considerable increase the estimate of the conditional probability for these rules (more than by δ, where δ is a threshold, for example $\delta = 0.2$), which means that achievement of this situation considerable increases the probability of achievement of a higher-level goal. The second condition says that different further actions are possible after this situation is achieved.

Thus, for each functional system FS, its set of rules PR is analyzed and new subgoals are identified. For each identified subgoal G_{new}, a new functional system FS_{new} which realizes this subgoal is generated at the lower (with respect to FS) hierarchical level. A set of patterns PR_{new} is created for the newly-generated functional system FS_{new} using the semantic probabilistic inference. To do this, the whole set of data on the animat X is examined in search for cases when the subgoal G_{new} was achieved, and the set of action estimates $d(t)$ of the functional system FS_{new} is calculated as described above. For all functional systems one level higher than FS_{new}, the set of their predicates is replenished with one more predicatePG_{new}, and new rules are generated. Thus the set of patterns for these functional systems is enriched with patterns that contain a new subgoal G_{new}.

7. The experiments

1. Experiment description. To study the control system described above, two experiments were performed differing in the task complexity level. A virtual world and an animat were modeled with the help of a computer program, and the main goal for the animat in both experiments was to pick special objects of the virtual world — "food". The animat has to learn how to find and pick the food efficiently.

The animat's world is a square field divided into cells which contains the following objects: empty cells ("grass"), obstacles ("obstacle"), and food ("food"). Objects-obstacles are located only along the virtual world's perimeter thus forming its natural borders. The animat may move across the field and perform 3 types of actions: to step one cell forward ("step"), to turn left ("left"), to turn right ("right").

In the first experiment, some amount of food is randomly distributed over the field. To pick food, the animat has only to turn to the cell containing food. When the animat steps to the cell with food, it is assumed that it "eats" the food, the cell is cleaned and new "food" object appears at random in another place of the field. Thus the amount of food in this virtual world is always constant.

The animat has nine sensors helping it to orient itself in the virtual world, eight of them are located around the animat: "in front of-to the left", "in front of", "in front of-to the right", "to the left", "to the right", "behind-to the left", "behind", "behind-to the right" and one is in its center: "center". Each sensor informs the animat about the type of object located in the corresponding cell and it may take the following values: "grass", "obstacle" or "food".

The second experiment is more complicated than the first one. In this experiment the virtual world has one more object conventionally called a "pill". Pills, like food, are randomly distributed over the field. Before eating food, the animat has first to find, pick and keep a pill. When it eats food, the pill disappears and, to eat the next portion of food, it again has to find and pick a pill, and so on. The pill is picked in the same way as the food is eaten: the animat has just to step to the cell containing the pill. However, if the animat has one pill, it cannot pick any more pills until it uses it to eat food. When the animat picks a pill, the cell is cleaned and a new pill appears in another place of the field at random, so the number of pills in this virtual world is always the same.

In the second experiment, the animat has ten sensors, nine of which are located in much the same way as the sensors in the first experiment and they get values "grass", "obstacle", "food" or "pill". One more sensor "pill availability" informs the animat if a pill is available and it gets the values "yes" or "no".

In order to evaluate efficiency of the control system we proposed, during our experiments we also made test comparison with systems built on the basis of Reinforcement Learning theory, described by R. Satton and E. Bartow [8].

For comparison, we have selected two control systems built on the basis of a popular algorithm for Reinforcement Learning — Q-Learning. The essence of this algorithm is in consequent refinement of estimates for the total reward value $Q(s_t, A_t)$ the system will obtain after performing an action A_t in a situation s_t, which is calculated by the formula:

$$Q^{(i+1)}(s_t, A_t) = Q^{(i)}(s_t, A_t) + \alpha(r_t + \gamma \max_A Q^{(i)}(s_{t+1}, A) - Q^{(i)}(s_t, A_t)).$$

The first of these two systems (Q-Lookup Table) is based on the table that contains Q-values for all possible situations and actions. Initially the table is filled in a random way. In the process of functioning, at each moment the system performs an action and refines the corresponding Q-values.

The second system (Q-Neural Net) uses an approximation of the function $Q(s_t, A_t)$ with the help of neural networks. In this case each possible action A_i uses a separate neural network NN_i. At each moment the system selects an action so that its neural network gives the highest estimate for Q-value, and after that the action is performed and the weights of the corresponding neural network are adapted.

Test comparison was made on a field 25 by 25 cells. The whole period of animat's functioning was divided into stages of 1 000 steps (cycles). We have evaluated and compared the amount of food the animat with different control systems could gather at each stage. It is evident that after the control system is fully trained and reaches its optimal behavior, the animat will start gathering approximately the same amount of food at each stage. Thus it is possible to evaluate both the efficiency of each control system on the whole and the speed of its learning.

2. Results of the first experiment. In the first experiment, the set of animat's predicates consists of twenty-seven sensor predicates – three predicates for each sensor: **(s = "grass")**, **(s = "obstacle")** and **(s = "food")**, and three activating predicates: **(A = "step")**, **(A = turn left")** and **(A = "turn right")**.

Initially the animat control system has only one functional system, and its goal is to find out if the food is available using its central sensor, and the corresponding goal-predicate is of the form **("center" = "food")**. When the animat reaches this goal, it is assumed that it "eats" the food.

In this experiment, the animat control system did not find any subgoals, which is quite natural, because in this task there are no situations satisfying the definition of a subgoal given above. The main aim of this experiment was to estimate working efficiency of a separate functional system.

Figure 5 shows the results of test comparison. The average values were calculated for each control system using the results of 20 tests. Each test lasted 50 000 steps. The amount of food on the field was kept constant and equaled 100.

From the diagram, it can be seen that the control system based on the semantic probabilistic inference is fully trained after 2 000 steps and it reaches its optimal behavior, i.e. it starts to react to all sensor data quite adequately.

The control system based on neural networks (Q-Neural Net) is learning more slowly and reaches its optimal behavior approximately after 40 000 steps.

Poor performance of the control system using Q-values table is explained by a large number of possible situations: if all three actions are taken into account, the number of different possible situations is 2 496. The results of our experiments show that, even after 50 000 working cycles, the control system

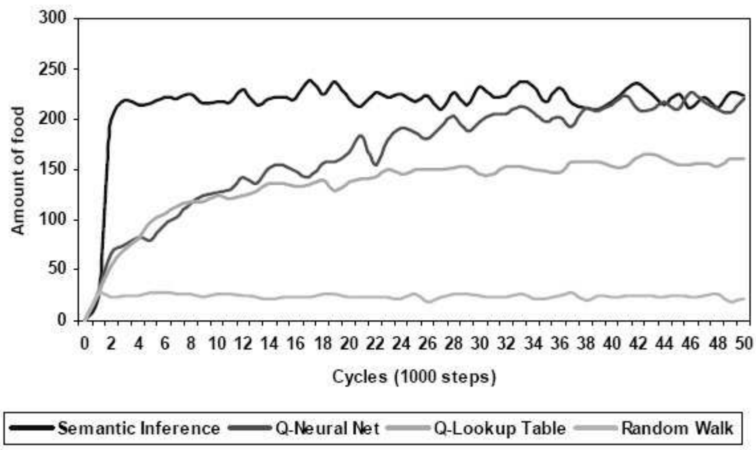

Figure 5. Amount of food collected by animats with different control systems in the first experiment

examines on average only about 1 450 situations. So, even after a long-term training, there occur situations when the system reacts inadequately.

The results of this experiment show that the semantic probabilistic inference allows a functional system to work efficiently enough and, at least, no worse than that on the basis of the Reinforcement Learning.

3. Results of the second experiment. This experiment differs fundamentally from the first one, because the task can be divided into two stages: first it is necessary to find a pill, then to find food. So, one of the main aims of this experiment was to demonstrate the possibility of automatic generation of the hierarchy of goals and results during the goal-seeking behavior.

Initially the set of animat's predicates consists of thirty-seven sensor predicates — four predicates for each sensor s that inform the animat about the conditions of the surrounding cells: **(s = "grass")**, **(s = "obstacle")**, **(s = "food")** and **(s = "pill")**, and one more predicate that informs if a pill is available **("available pill" = yes)**. There are also three activating predicates: **(A = "step")**, **(A = turn left")** and **(A = "turn right")**.

At the beginning, the animat control system has only a basic functional system, and its goal is to achieve the situation when, at the same time, a pill is available and the central sensor indicates that food is found. The corresponding goal-predicate is of the form $PG_0 =$ **("center" = "food" AND "available pill" = yes)**. When the animat achieves this goal, it is assumed that it "eats" food.

At each test started during our experiment, the animat control system did stably identify a new subgoal described by the subgoal-predicate $PG_1 =$ **("available pill" = yes)** and created the corresponding functional system

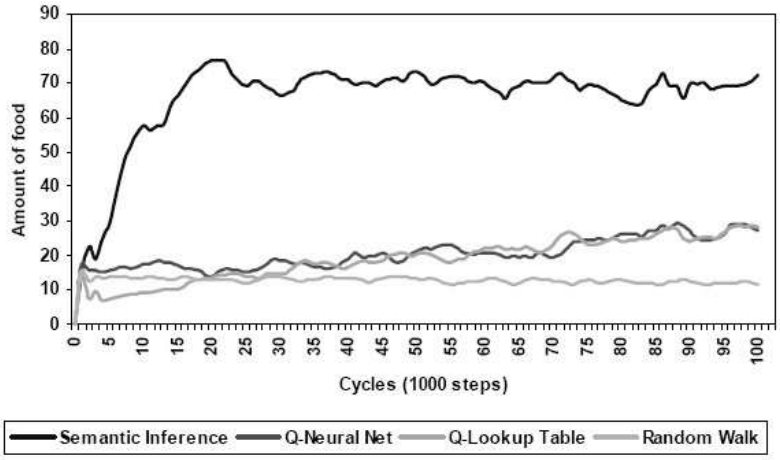

Figure 6. Amount of food collected by animats with different control systems in the second experiment

for it. The control system was working in the following way. When the animat had no pill, the pattern $PG_1 \to PG_0$ was launched as the most probable one in this situation. It passed the control functions to a lower-level functional system that performed the pill search. And when the animat had a pill, rules with a higher probability were launched in the basic functional system and, as a result, food was found.

The results of this experiment are shown in Figure 6. The diagram shows the average values for every control system based on 20 tests. In each test the animat was given 100 000 steps, and during this time the animat had to learn how to accomplish the given task efficiently. The amounts of pills and food on the field were kept constant: 100 objects of each type.

As it can be seen from the diagram, the control system based on the semantic probabilistic inference surpasses the Reinforcement Learning systems both in the speed of learning and in the quality of its functioning.

The control system based on neural networks (Q-Neural Net) has shown in this experiment a poor learning ability and unstable functioning. The main problem was related to the fact that it could not learn to stably and adequately react to sensor data about pill availability and often passes the pills by, even after 100 000 steps of training.

Additional experiments have shown that the control system based on neural networks (Q-Neural Net) is in some cases capable to learn to react to all sensor data correctly when the training period is increased up to 300 000–500 000 steps. But in our opinion such a long learning time is unacceptable for an adaptive system.

The control system based on the Q-values table could not reach optimal behavior even after 500 000 steps. In many respects it is related to the great number of possible situations: when solving this task, the animat can face 137 538 different situations.

8. Conclusion

The results of our experiments show that, when the environment is getting more and more complicated, the ability to identify and achieve subgoals becomes the fundamental one for efficient achievement of a final goal. Even though this model of an adaptive control system uses a rather simple method of subgoal identification, this possibility provides considerable advantages in learning. As it can be seen from the experiment, the use of functional systems hierarchy and of the algorithm for subgoals identification allows the proposed control system to learn and accomplish the given task quite efficiently. The existing approaches based on neural networks and Reinforcement Learning are unable to identify subgoals automatically and thus are inferior in more complicated experiments.

References

[1] Vityaev E.E. Purposefulness as the principle of brain activity // Computational Systems. — Novosibirsk, 1997. — Vol. 158. Models of cognitive process. — P. 9–52. (in Russian).

[2] Vityaev E.E. Probabilistic forecasting and prediction as principle of brain activity // Computational Systems. — Novosibirsk, 1998. — Vol. 162. Measurement and models of cognitive processes. — P. 14–40 (in Russian).

[3] Vityaev E.E. Formal model of brain activity founded on prediction principle Computational Systems. — Novosibirsk, 1998. — Vol. 164. Models of cognitive process. — P. 3–62 (in Russian).

[4] Mikhienko E.V., Vityaev E.E. Functional System Operation Modeling // Proc. of the All-Russian Conf. "Neuroinformatics-2004". Part 2. — Moscow: MEPHI, 2004. — P. 124–129 (in Russian).

[5] Vityaev E.E. N.A.Bernshtein motions theory explanation // Proc. of the All-Russian Conf. "Neuroinformatics-2005". Part 1. — Moscow: MEPHI, 2005. — P. 234–240 (in Russian).

[6] Vityaev E.E. Semantic approach to knowledge base creating. Semantic probabilistic inference of the best for prediction PROLOG-programs by a probability model of data // Logic and Semantic Programming. — Computational Systems. — Novosibirsk, 1992. — Vol. 146. — P. 19–49 (in Russian).

[7] Vityaev E.E. Method of regularities determination and method of prediction // Computational Systems. — Novosibirsk, 1976. — Vol. 67. Empirical Prediction and Pattern Recognition. — P. 54-68 (in Russian).

[8] Sutton R. and Barto A. Reinforcement Learning: An Introduction. — Cambridge: MIT Press, 1998. — Available at
`http://www-anw.cs.umass.edu/~rich/book/the-book.html`

[9] Anokhin P.K. Biology and neurophysiology of the conditioned reflex and its role in adaptive behavior. — Oxford: Pergamon press, 1974. — 574 p.

[10] Anokhin P.K. Functional system // Dictionary of behavioral science. — New York, 1973. — P. 153.