Implementing Cognitive Architecture Plan-Image to Search Tasks in Agent Models

Alrksandra Mashkova
Orel State University
named after I.S. Turgenev
Orel, Russia
CEMI RAS
Moscow, Russia
aleks.savina@gmail.com

Olga Savina
Orel State University
named after I.S. Turgenev
Orel, Russia
Belgorod State University
Belgorod, Russia
o.a.savina@gmail.com

Eugene Mashkov
Orel State University
named after I.S. Turgenev
Orel, Russia
ppsnnt@gmail.com

Abstract—In the article we discuss some questions connected with cognitive structures and behavior algorithms of an artificial social agent. We suggest a cognitive Plan-Image architecture that is based on the concepts of TOTE model by Miller, Galanter and Pribram. In order to evaluate efficiency of the architecture we use a classical search task on a two-dimensional grid. Each agent has a map that reflects his current information about the location of objects (Image) and alternative search algorithms (Plans). Objects in the grid may be arranged randomly or according to a certain law even or centered pattern. The experiment aims to assess how effectiveness of the search changes depending on the amount of available information and agents search algorithms.

Index Terms—artificial agent, cognitive architecture, agent model, TOTE model

I. Introduction

The idea of autonomous agents has emerged in the development of artificial intelligence systems. The concept of multi-agent system (MAS) assumes that its members (agents) are able to act independently and cooperate with each other. As the scope of multi-agent systems is very wide, there are also a sufficient number of types of the agent's architectures, based on different concepts. There are deliberative agent architectures, reactive architectures and multi-level architectures [1]. The examples of deliberative agent architectures are ACT-R and SOAR [2], [3], [4]. The sample of reactive architecture is STRIPS system [5]. Architecture of artificial agents based on genetic algorithms of J. Holland may be outlined separately [6] as well as architecture based on neural network [7], [8]. These approaches may be combined. For example, in the CLARION architecture a two-stage decision-making model is adopted [9]. The lower level is a neural network, which distinguishes the most important information from the array of input data, and the upper level is a deductive system.

Regardless of mentioned studies the concept of agent-based modeling (ABM) was proposed, a vivid example of which is the sugar model [10]. This direction has developed in the methodology of artificial societies and is intended to reproduce the dynamics of complex social and economic systems through the behavior of individual actors-agents [11].

Despite significant differences of multi-agent approach to artificial intelligence and agent-based social modeling, they have a number of common features: reproduction of decision-making process by computer algorithms, the ability of agents to interact with each other and with the environment. An important task in the development of modern agent-oriented models is integration of knowledge about human cognitive processes, opportunities for their formalization and computer implementation with certain social structures, sufficient to create a decision making environment [12]. Relation between the fields of ABM and MAS has been broadly discussed in the recent decade [13], [14], [15], and the overall conclusion is that both directions would benefit from their interaction [16]. The direct use of well-known cognitive architectures in agent-based models is difficult for several reasons. Firstly, the existing architectures are aimed at solving technical problems, and not at the reproduction of social behavior. One of the exceptions in this regard is use of CLARION architecture for modeling agent's survival strategies in simple communities [17]. Another successful exception is implementing BDI architecture to simulating land use in a village [15]. Steps to create standardized models of decision-making agent solutions are presented in [18]. Secondly, the implementation of a complex architecture requires significant computing power. In technical systems, as a rule, several tens to several hundreds of agents cooperate, in social simulation tasks thousands of agents are involved [19], [20]. Using a multi-level architecture limits this number. In the above example with CLARION architecture [17] modeling was performed for 30 agents, which is quite far from the number of individuals in real societies. However, a simplified BDI architecture, integrated into the GAMA modeling platform, shows good computational results [15].

Due to these reasons we consider the architecture Plan-Image for an artificial social agent, which is a simple computable architecture for modeling decision making process in an artificial society. The architecture got its name according to the basic concepts of TOTE cognitive model (Test-Operate-

Test-Exit), proposed by D. Miller, J. Galenter and K. Pribram [21]. The advantages of this model are simplicity of software implementation and use of semantic networks for the presentation of information available to the agent.

Plan-Image architecture takes into account the limited rationality of agents. Rationality limitations can have a cognitive nature, i.e. the limitations of the available information to the agent and its computing power [22], or the deeper causes, due to unconscious processes [23]. TOTE model terms are convenient for presenting agents cognitive limitations.

To evaluate the effectiveness of the architecture we set a task for the agents to find the objects on the map. Through computer experiments we examine the influence of the agent's presentations of the environment on the search speed in conditions of incompleteness of initial data.

II. COGNITIVE ARCHITECTURE PLAN-IMAGE

At the heart of the proposed cognitive architecture of an artificial agent is TOTE model of behavior, the basic concepts of which are Plan and Image [21]. Plan is a hierarchically built process in the body that can control the order in which any sequence of operations should take place. When a person performs a Plan, he does it step by step, completing one part and moving to the next. For the agent Plan corresponds to the algorithm that determines its behavior in the environment, such as a search algorithm. Image is the accumulated and organized knowledge of a man about himself and about the world in which he exists. Image includes all the evaluation and the facts acquired and organized by the person with the help of those concepts, images and attitudes that he was able to develop. Image of the agent is a set of data available to it. Formulation of Plans by the agent should include the part of the Image; Image should be a part of the Plan, as otherwise Plan cannot serve as a basis to guide behavior. You can make changes to Images just by making Plans for collecting, accumulation and processing of information. Changes in Plans are made only on the basis of information from Images. Test-Operate-Test-Exit scheme (see Fig. 1) is a concise description of the link between the Image and the performed actions of the Plan [21]. Test phase includes definition of knowledge required for the comparison and Operate phase actions that the body performs in accordance with the acquired knowledge. Exit phase corresponds to the achievement of the goal. Planning can be viewed as a list of samples to be performed. When there is a clear Image of the desired result, it can be used to create the conditions to be tested, and being organized in the correct sequence, these tests make the strategy of the possible Plan.
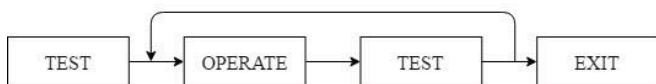

Fig. 1. Scheme of TOTE model

In a search task the significant Images are the following:
1. Objective World Image representing a map of the world with agents and other objects located on it.

2. Subjective World Image, inherent in a certain agent, is reflected in the model through personal maps.

The change of Images occurs according to the implementation of the agents Plan. Agent search algorithm works on the basis of its Subjective World Image, which the agent builds on the received information.

III. DESCRIPTION OF THE MODEL

Let's formulate the search task for the evaluation of the proposed architecture. Model environment is presented by a map. Objects on the map (fruit trees) are placed in accordance with a certain law. The cell can be empty or contain one or more trees. There are two different patterns of tree location: even and centered. Examples of the patterns are shown in Fig. 2, where painted cells represent trees and a number in the cell represents number of the trees in it.
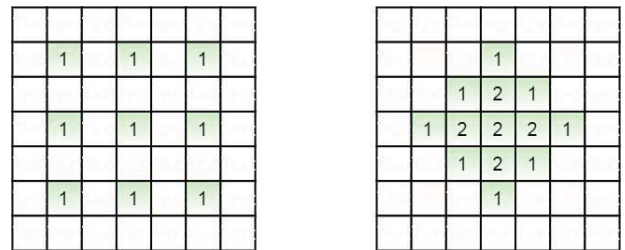

Fig. 2. Patterns of objects location on the grid a) even b) centered

Agents are generated at random locations on the map and live for a predetermined number of cycles. We believe that the number of fruit trees equals the number of agents. The purpose of agent's life is to find a personal fruit tree. When the agent finds a free tree, it remains on the site until death. Each agent may use fruit of only one tree; each tree is able to feed only one agent (link 1:1). Information is available for agent only about those cells, which it has visited. The agent can move to one of the neighboring cells (up / down / left / right / diagonally) at once. Having been in the cell, the agent can learn that a) it is empty, b) it contains a number of trees, but they are busy, or c) it has free trees. If there are trees available, the agent takes one of them. Otherwise, it puts the received information on its map and continues to search. Each agent has several (at least two) patterns: even, centered, etc. To fill the map according to the even pattern it is enough to know the coordinates of any filled cell, and a step in which one filled cell is separated from the other. To fill the map according to the centered pattern, it should know the radius of the filled zone. The pattern is set by default, when the agent starts searching. Let's formulate the task in terms of Plan-Image architecture. Model environment is Objective World Image. Each agent has the original Plan and Subjective World Image. Components of the Subjective World Image of agent are an environment map (a blank grid), coordinates of its location on the grid and patterns of possible locations of the trees. The agent's Plan consists of rules for filling the map patterns and obtaining information, laws of motion on the grid (depending on the pattern) and a search algorithm for finding the shortest route to the desired location. In the search task Test phase corresponds to checking the

presence of an object in the current cell, Action phase - the transition to the next available cell. Exit status is reached by agent, if it has found a free tree and has taken it. At each modeling tact the agents, which are not in Exit phase, perform Test phase. If Test is successful, i.e. agent found a free object, agent goes into Exit. Otherwise, it performs Action and repeats the cycle.

IV. ALGORITHMS OF THE MODEL

Modeling dynamics of the listed indicators as a result of events is crucial for estimating efficiency of the Program in general. As an ultimate goal of the research is increase of level of proficiency in Russian among population of the Russian Federation and increase of number of people who speak Russian in the CIS, Baltic countries and other foreign countries, we should divide agents into clusters so that each cluster was referred by the set of the relevant events of the Program (table 1).

In general, the agent map can be represented as a matrix, in which the following items are marked: free cells, visited cells, found trees and supposed trees. We use the following notations of the described parameters (Fig. 2):

1. *Start* – the cell from which the agent starts its search;
2. *End* – the cell in which the agent is at the end of modeling time;
3. $N = 0$ – the visited cell was empty;
4. $N = num, num > 0$ – the visited cell contained trees, but all of them were occupied;
5. $M = 0$ – a cell that is supposed to be empty according to the pattern;
6. $M = num, num > 0$ – a cell that is supposed to contain trees.

The algorithm of the model work consists of the following steps [24]:

1. Set the pattern by default (even) and the law of motion in lines.
2. Find the nearest filled cell moving in accordance with a given law of motion. Put down information about the visited cells on the map. If more than 1/4 of the map has been completed, go to step 3 in the search process, otherwise go to step 4.
3. Change the pattern into centered and set the law of motion to the center.
4. Move according to a given law of motion to the next filled cell. Put down the information about the visited cells on the map.
5. Determine the step (the distance between two filled cells) and form the map according to the current pattern.
6. Find the shortest route to the nearest filled cell according to the resulting map. Move across the grid and put down information about the visited cells on the map. If the information is received, corresponding centered pattern,

then proceed to step 3. If the received information is contrary to the existing map, go to step 5. Information may be considered controversial, if the cell is indicated as empty on the map, but in reality there are trees, or the map shows the trees while the cell is empty.
7. Repeat step 6 until you find a free tree or until the agent is dead.

Figure 3 shows how the agent's map changes during the search process. The agent is created at the bottom left corner, which is an empty cell. The agent moves according to his default low of motion until he finds a filled cell. The cell contains one tree and it is already occupied, so the agent puts this information on its map (step 2). The agent continues its search until it finds the next filled cell. Now it is able to calculate the distance between the filled cells and represent it on its map (step 5). Then the agent finds the shortest way to the nearest filled cell and occupies a tree in it (step 7).
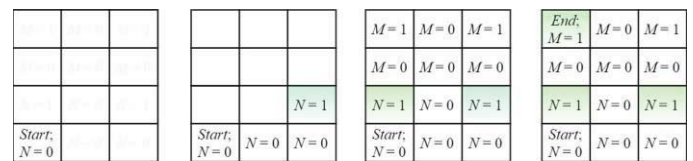

Fig. 3. Agent's map during the search process (from left to right): start, step 2, step 5, step 7

Let's consider the search algorithms that are used to find the way to the target cell for the law of motion along the lines:

1. If the line is passed, then go to step 2. Otherwise: choose the nearest freeedge (left or right) and select the target free cell in the current line, the closest one to the selected edge.
2. If you are in at the bottom or at the top line, then go to step 3. Otherwise: choose the nearest free vertical edge of the map (top or bottom) and choose the target cell in the current column that is the closest to the selected edge. 3. Select the closest empty cell in the current column.

Search algorithm for the target cell in the case of motion law to the center is analogous, only in step 2 the cell in the direction of the center of the map is selected.

V. EXPERIMENTAL STUDIES

As a software tool for the implementation of the algorithm, agent-based modeling environment NetLogo has been chosen [25]. The initial parameters of the model are:

- type of pattern,
- low of motion,
- degree of uncertainty,
- starting value of the agent's energy
- initial coordinates of the axes to generate the even pattern,
- offset axes for the even pattern,
- number of trees in the node for the even pattern,
- center coordinate of axes for the centered pattern,
- radius of the centered pattern,

– number of trees in the central cell of the centered pattern.

When launching the model we generate agents and there pattern; then a modeling cycle is started. The purpose of the experiment is to evaluate how search efficiency changes depending on the information about the environment, available for the agents. In general, there are four options for the search organization, depending on the initial information:

1. Maximum certainty: agents have full information about the original distribution of trees and other agents. Search of each agent is the most effective and is based on the assumption that other agents are also looking for the shortest route.
2. High certainty: agents have information about the original distribution of trees and are looking for the shortest route to the nearest tree. In this case the following situation is possible: one agent is coming to the tree, another is first to take it.
3. Medium certainty: agents have information on the size of the grid, possible patterns and placement of trees.
4. Low definition: agents have no information, looking for blindly.

In the first case, the task has absolute optimum on total search time on the map, which can be found by using the following algorithm:

1. Find the nearest tree for each agent and assign it as target.
2. Check all the trees and determine the presence of conflict that occurs when one tree is the nearest to several agents. When there is a conflict that agent wins whose route to the target tree is shorter. If two or more agents can achieve the same tree for the same number of steps, then check the availability of other trees for conflicting agents and give the victory to the agent, whose route to the next tree would be longer. If all the conditions have the same value, determine the winner at random. Appoint the next nearest tree as target for other agents.
3. If step 2 had no conflict, minimum total time will be maximum search time for an agent on the map. If there were conflicts, it is necessary to repeat step 2, excluding from consideration the trees which are not the subject of the conflict. At each repetition of step 2 at least one tree is excluded from consideration.

For agent various ways of movement on the map are possible: random, vertically and horizontally. These methods are used when searching for low-certainty, and in the primary search at the medium level of uncertainty. At the end of the modeling some statistics is presented, including average search time of a free tree and the percentage of agents, the search for which was unsuccessful (failed agents). The values of these parameters are compared for the cases of different definitions (2, 3, 4) and the conclusion on the impact of patterns on search success is made. In case 3 it is also estimated how many agents have been successful, having the wrong pattern, by pure chance.

After holding a series of experiments for each level of uncertainty, the type of pattern and movement pattern data is listed in the summary table 1 for statistical comparison [26]. The set time of modeling is 100 clock cycles, the grid size is 50*50 cells, the initial generation is 100 agents, the number of trees is equal to the number of agents. Let's explain the collected statistics. To calculate statistics, we use the following equations:

$$Tr_{Av} = (Tr_F + Tr_S - A_S)/A, \qquad (1)$$

$$R_{Av} = C/A, \qquad (2)$$

$$A_F = (A - A_S)/A, \qquad (3)$$

where A - the total number of agents; A_S - number of successful agents; A_F percentage of failed agents; TR_S - number of trees found by successful agents; Tr_F - number of trees found by failed agents; C - number of cells passed by all agents; Tr_{Av} - average number of found trees; R_{Av} - average route of the agents.

TABLE I. MAIN CHARACTERISTICS OF EFFICIENCY OF AGENTS' SEARCH.

Certainty Parameters	Low	Medium	High
Even Pattern			
Tr_{Av}	1,6	2,3	3,1
R_{Av}	34,8	18,6	12,8
A_F	35	13	6
p	–	50,9	–
Centered Pattern			
Tr_{Av}	0,2	0,4	9,5
R_{Av}	91,1	49,1	16,7
A_F	55,8	22,1	0
p	–	73,5	–

The agent is considered successful, if it was able to find and occupy the tree. When calculating Tr_{Av} only those objects-trees are taken into account, which the agents met on their route to the cell where they stopped. This parameter shows the average number of trees, found by agents in the course of their travel. R_{Av} shows the average route that is needed to find a free tree; parameter p (percentage of successful agents with precise pattern) reflects how many successful agents found the trees, having correct information about the map (in the case of medium certainty).

As it follows from the obtained data, agents whose Subjective World Image contains detailed information about the map were the most successful. With this information they reached their goal quickly and effectively, but in the real-world tasks such a high degree of certainty is very rare.

The least effective model shows itself at low certainty when Image contains only a blank map. In this case, agents spend more time wandering around empty cells. Compared

with medium degree of uncertainty, the length of the route for the normal pattern differs slightly, but for the centered pattern these values differ greatly, since agents, appeared far from the center of the pattern, spend much of their time on the way out of an empty part of the field.

The model with an average level of uncertainty is the most appropriate to real conditions. In this case, agents have the original pattern of objects location and can change it to a more appropriate one, if meet discrepancy between the actual position of objects on the grid and position expected by current pattern. Studies of this model show that number of agents who have reached their goal using a pattern is significantly more than 50 percents of the total number of agents, and these differences are more obvious for centered pattern, in which the blind search provides inefficient results.

VI. Conclusion

In the article the results of the work are presented, which continue the direction of integration of cognitive modeling and social simulation, pledged in the writings of R. Sun [9], N. Gilbert [13], P. Caillou [15], M. Wellman [16] et al. At this stage of research, our purpose was to formalize cognitive limits of rationality of an artificial agent. The cognitive model TOTE was selected as the basis of the agent architecture. Analysis of experimental data has shown that the agents behavior with Plan-Image architecture differs significantly from both the agents with blind search procedures and the agents with the perfect rationality. Since the problems of the real world are characterized by inconsistency and inaccuracy of information, Plan-Image architecture can be used to simulate the behavior of agents in more difficult conditions than a simple two-dimensional network. For example, when looking for a job or taking a decision on migration, behavior of agents with Plan-Image architecture is much closer to reality than using common assumptions about the full availability of information [26].

Acknowledgment

References

[1] Gorodetsky, V.I., Grushinsky, M.S., Khabalov, A.V. "Multi-agent systems (review)," News of artificial intelligence, vol.2, pp. 64-116, 1998.

[2] J. Anderson, The architecture of cognition, Cambridge, MA: Harvard University Press, 1983.

[3] N. Taatgen, C. Lebiere, J. Anderson, "Modeling Paradigms in ACT-R," in: Cognition and Multi-Agent Interaction from Cognitive Modeling to Social Simulation, Cambridge: Cambridge University Press. pp. 29-52, 2006.

[4] A. Newell, Unified theories of cognition, Cambridge, MA: Harvard University Press, 1990.

[5] R.E. Fikes, N. Nilsson, "STRIPS: A new Approach to the Application of Theorem Proving to Problem Solving," Artificial Intelligence, vol.5(2), pp. 189-208, 1971.

[6] J.H. Holland, J.H. Miller, "Artificial Adaptive Agents in Economic Theory," AEA Papers and Proceedings, vol.5, pp. 365-370, 1991.

[7] A.A. Zhdanov, "Autonomous artificial intelligence," Moscow, Russia: BINOM. Knowledge Laboratory, 2009.

[8] Z.B. Sokhova, Z.B. Shikzatov, "A model of cooperated agent guards with the needs and motivations," in: Scientific session of the MIPI 2013.

Part 2: XV Russian Scientific and Technical Conference Neuroinformatics, vol.2, pp. 274-281, Moscow, Russia: MIPI, 2013.

[9] R. Sun, "Prolegomena to Integrating Cognitive Modeling and Social Simulation," in: Cognition and Multi-Agent Interaction From Cognitive Modeling to Social Simulation, Cambridge: Cambridge University Press, pp. 3-28, 2006a.

[10] J. Epstein, R. Axtell, Growing Artificial Societies: Social Science From the Bottom Up, MIT Press, Brookings Institution, 1996.

[11] S. Moss, "Relevance, realism and rigour: A third way for social and economic research," CPM Report No. 99-56. Center for Policy Analysis, Manchester Metropolitan University, (1999).

[12] R. Sun, "The CLARION Cognitive Architecture: Extending Cognitive Modeling to Social Simulation," in: Cognition and Multi-Agent Interaction from Cognitive Modeling to Social Simulation, Cambridge University Press, pp. 79-102, 2006.

[13] Gilbert, N.: When Does Social Simulation Need Cognitive Models? In: Cognition and Multi-Agent Interaction From Cognitive Modeling to Social Simulation. Cambridge University Press, P. 428-432 (2006)

[14] M. Niazi, A. Hussain, "Agent-based computing from multi-agent systems to agent-based models: A visual survey," Scientometrics, vol.89, pp. 479-499, 2011.

[15] P. Caillou, B. Gaudou, A. Grignard, C.Q. Truong, P. Taillandier, "A Simpleto-use BDI architecture for Agent-based Modeling and Simulation," in: The Eleventh Conference of the European Social Simulation Association (ESSA 2015), Sep 2015, Groningen, Netherlands, 2015.

[16] M.P. Wellman, "Putting the agent in agent-based modeling," Autonomous Agents and Multi-Agent Systems, November 2016, vol.30(6), pp.1175-1189, 2016.

[17] R. Sun, I. Naveh, "Social institution, cognition, and survival: a cognitive-social simulation," Mind and Society, vol.6, pp.115-142, 2007.

[18] B. Muller, F. Bohn, G. Dreler, J. Groeneveld, C. Klassert, R. Martin, et al., "Describing human decisions in agent-based models ODD+D, an extension of the ODD protocol," Environmental Modelling and Software, vol.48, pp.37-48, 2013.

[19] P. Froncek, "The Effects of Network Structure on the Emergence of Norms in Adaptive Populations," Journal of Artificial Societies and Social Simulation, vol.18 (4), p.14, 2015.

[20] F. Santosa, D.F. Adamattib, H. Rodriguesa, G. Dimuroc, E. Jerezc, G.A. Dimuroa, "Multiagent-Based Tool for the Simulation of Social Production and Management Processes of Urban Ecosystems Using the JaCaMo Framework: A Case Study of San Jernimo Vegetable Garden Seville, Spain," Journal of Artificial Societies and Social Simulation, vol. 19(3), p.12, 2016.

[21] G.A. Miller, E. Galanter, K.H. Pribram, Plans and the structure of behavior, New York: Henry Holt, 1960.

[22] H. Simon, Models of Bounded Rationality. Vol. 1, MIT Press, 1982.

[23] E. Bignetti, "The functional role of free-will illusion in cognition: The Bignetti Model," Cognitive Systems Research, vol. 31-32, pp.45-60, 2014.

[24] A.L. Mashkova, N.S. Kononov, "The algorithmic implementation of the cognitive limitations of rationality of artificial agents in computer experimental models," in: International scientific and practical Internet-conference Information systems and technologies 2015. http://youconf.ru/isit2015/materials/manager/view/428. Last accessed 4 Mar 2018

[25] L. R. Izquierdoa, D. Olarub, S. S. Izquierdoc, S. Purchaseb, G.N. Soutarb, "Fuzzy Logic for Social Simulation Using NetLogo," Journal of Artificial Societies and Social Simulation, vol.18 (4), pp.1-15, 2015.

[26] Angus, S. D., Hassani-Mahmooei, B.: Anarchy Reigns: A Quantitative Analysisof Agent-Based Modelling Publication Practices in JASSS, 2001-2012, Journal of Artificial Societies and Social Simulation, vol.18 (4), p.16, 2015.