Cogn Process (2009) 10:95–99
DOI 10.1007/s10339-008-0236-9

LETTER TO THE EDITOR

# ACACIA: an agent-based program for simulating behavior to reach long-term goals

**Francesc S. Beltran · Vicenç Quera · Elisabetta Zibetti · Charles Tijus · Meritxell Miñano**

Received: 22 January 2008 / Revised: 30 September 2008 / Accepted: 2 October 2008 / Published online: 28 October 2008


**Abstract** We present ACACIA, an agent-based program implemented in Java StarLogo 2.0 that simulates a two-dimensional microworld populated by agents, obstacles and goals. Our program simulates how agents can reach long-term goals by following sensorial-motor couplings (SMCs) that control how the agents interact with their environment and other agents through a process of local categorization. Thus, while acting in accordance with this set of SMCs, the agents reach their goals through the emergence of global behaviors. This agent-based simulation program would allow us to understand some psychological processes such as planning behavior from the point of view that the complexity of these processes is the result of agent–environment interaction.

**Keywords** Agent-based simulation · Long-term goals · Local categorization

## Introduction

In the last years, some approaches state that behavior emerges from the interaction of the organism with the environment (Bakker 2000; Brooks 1999; Holland 1995; Maes 1997; Meyer and Guillot 1991). In fact, many complex global behavioral events emerge from decentralized, independent components that interact among them and with the local environment. Some examples include traffic jams (Resnick 1994), coordinated motion group such as bird flocking (Reynolds 1993), herds (Werner and Dyer 1992), pedestrian behavior (Schreckenberg and Sharma 2002), and robots collecting objects (Maris and te Boekhorst 1996). In all those systems, a set of local rules is organized in terms of the actions to be performed in order to respond to the circumstances of the immediate environment. These local rules, defined as sensory-motor couplings (SMCs) by Braitenberg (1984), guide the organism–environment interaction.

The aim of the agent-based simulation approach is to emulate the behavior of natural organisms in complex, dynamic environments. By creating an artificial agent able to perform certain behaviors in a virtual environment, it is possible to try to determine the internal mechanisms underlying these behaviors. We present ACACIA, an agent-based simulation program that simulates a multi-agent system where agents interact with their environment and other agents in order to reach long-term goals (Zibetti et al. 2001a), which are defined here as places that are desirable for the agents and that may be some distance away. The program shows how the agents can reach a long-term goal based on a set of SMCs that controls the agent's local interaction with its environment and with other agents. The set of SMCs does not specify a global internal representation of the environment or a sequence of steps necessary to reach the goal; rather, it is a process of local categorization that determines how the agent relates locally with objects and other agents in its environment (Zibetti et al. 2001b).

F. S. Beltran · V. Quera · M. Miñano
Adaptive Behavior and Interaction Research Group,
Universitat de Barcelona, Barcelona, Spain

E. Zibetti · C. Tijus
Laboratoire de Cognitions Humaine et Artificielle,
Université de Paris-8, Paris, France

F. S. Beltran (✉)
Departament de Metodologia de les Ciències del Comportament,
Facultat de Psicologia, Universitat de Barcelona,
Campus Mundet, 08035 Barcelona, Spain
e-mail: francesc.salvador@ub.edu

Thus, through the SMCs the agent groups and differentiates the entities in its environment based both on their physical properties and on the task the agent must perform. In the next sections, we show how the program works, some previous results using the program and an example of a simulation experiment that illustrates ACACIA's abilities.

## The ACACIA program

ACACIA is implemented in StarLogo (Colella et al. 2001; Resnick 1994), a programmable environment designed to model multi-agent simulation systems. It was developed using the Java StarLogo 2.0 version, which runs on different operating systems, including Windows, Mac OS and Linux. The program simulates a discrete, two-dimensional microworld that can be either a torus or a closed space surrounded by walls. In both cases, the surface is divided into 50 × 50 square cells (or patches). The microworld contains three different kinds of entities: goals, obstacles, and agents (see Fig. 1).

### Goals

Goals are static entities that are sought by agents. Goals are shown red on the screen and each one occupies one patch, or location. When the simulation starts, goals are scattered randomly throughout the microworld. The number of goals can vary from 1 to 20 and is set by the user.

### Obstacles

Obstacles are areas composed of many contiguous patches that cannot be occupied by agents and which agents cannot see through. Obstacles are shown yellow on the screen, and are randomly distributed throughout the microworld, their number (0–20) and shape (regular/irregular) being set by the user.

### Agents

Agents have two-dimensional coordinates that specify their positions at time $t$, and headings that indicate the directions of their movements. An agent can move one cell or patch per time unit in any direction relative to its current position. Agents scan their neighborhood in order to identify different kinds of entities (goals, obstacles, and other agents) that they might encounter while exploring the environment. This mechanism has three parameters (a) neighborhood radius or depth of the agent's field of perception; (b) neighborhood angle or width of the agent's field of perception; and (c) scan resolution, which specifies how precise the agent's perception is within its neighborhood. The higher the resolution the greater the number of patches the agent can scan. The three parameters define a dynamic perceptual field in front of each agent so that only goals, other agents, and obstacles that lie in that field can be currently perceived by it.

Depending on the entities currently detected by an agent, its "internal" status can change (agent statuses are represented by different colors on the computer screen, which can be detected by other agents). The initial status

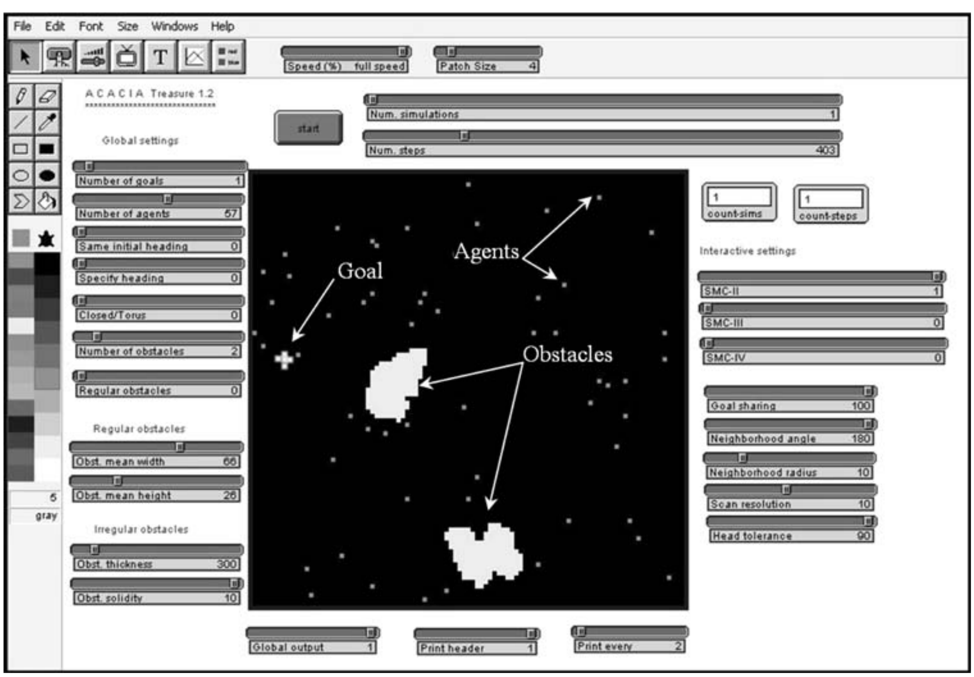


**Fig. 1** The ACACIA screen, displaying the entities in the microworld and the sliders that allow the user to manipulate the simulation parameters

for all the agents is "explorer", as they explore the environment looking for goals. When an agent reaches a goal, its status changes to "rich"; rich agents do not react to other entities; instead, they keep moving straight ahead in a random direction from the patch where the previous goal was located until they encounter an obstacle or a wall, then they disappear from the microworld because they have already reached the goal. The reason why rich agents do not disappear as soon as they reach the goal is that we felt this new information could be used by the other agents to reach it as well. If different entities are simultaneously detected in the neighborhood, then goals have priority over agents, and agents have priority over obstacles.

An explorer agent has five hierarchical SMCs that allow it to respond differently to the entities in the environment: (a) *SMC-I* If an agent is exploring the environment and it detects a goal, then the agent sets the coordinates of the goal as its target location, and moves one patch forward. (b) *SMC-II* If an agent is exploring and it detects a rich agent in its neighborhood, then the former sets its heading opposite to that of the rich agent, then it moves one patch forward; moving in a direction opposite to that of a rich agent may be a successful behavior because the rich agent is coming from a goal. (c) *SMC-III* If an agent is exploring and detects another explorer agent in its neighborhood, then the former first checks whether their headings are similar (within a tolerance limit defined by the user) and, if so, it sets its new heading so that it is the same as that of the latter, and then moves one patch forward; moving in the same direction as another explorer agent that is ahead may be a successful behavior because the latter might have already seen a goal and be heading toward it. (d) *SMC-IV* If an obstacle or a wall is detected, the agent first checks whether it had already detected an obstacle or a wall in the previous simulation step. If not so, it changes its heading 90° randomly to the left or to the right; otherwise the agent changes its heading 90° to the left if it had previously changed it to the left or 90° to the right if it had previously changed it to the right; it then moves one patch forward, provided there is no obstacle or wall in the patch to be occupied. Finally, (e) *SMC-V* If no entity is detected, the agent simply moves forward one patch following its heading and continues exploring.

The user can selectively set on or off SMCs II, III and IV for all explorer agents while the simulation is running. If the three SMCs are set on and an explorer agent simultaneously meets a rich agent, an explorer agent and an obstacle, SMC-II has priority over SMC-III and -IV.

## Running ACACIA

When the program is run, two StarLogo windows are displayed on the screen. One contains the procedures coded in StarLogo language and the other shows the ACACIA virtual microworld. Buttons and sliders for controlling simulation parameters are also shown (see Fig. 1). Other windows are: (a) an output window that shows a summary of the parameters and the values of some specific dependent variables as the simulation progresses (e.g., the percentage of agents reaching the goal, the percentage of agents acting in accordance with SMC-III, and so on); (b) a graphical window showing those dependent variables as time series; and (c) an information window that provides details of the main features of the simulator and how it works.

The simulation dynamics vary depending on the microworld parameters. In previous experiments, we found that when the agents' perceptual ability was limited by defining a neighborhood radius equal to ten patches, setting SMC-III on enabled agents to reach the goal. Moreover, a collective searching behavior emerged, whereby agents followed each other. Note that no specific SMC for the agents defined such collective searching behavior (Miñano and Beltran 2004). We also found that SMC-II (whereby agents headed in the opposite direction when they met rich agents) also increased the probability of reaching the goal. As more agents reached the goal, there were more opportunities for the other agents to see them and to act in accordance with SMC-II. Thus, the activation of SMC-II produced an emergent global migration of the agents to the goal. Nevertheless, this collective migration was not specified in an SMC (Miñano and Zibetti 2005).

In summary, we observed that the limited perceptual features of the agents were compensated by a collective behavior emerging from the activation of SMC-II and/or SMC-III. Like other computer models (Couzin et al. 2002; Epstein and Axtell 1996; Hemelrijk 1996, 2003), local interaction among agents results in cognitive optimization, i.e. "collaboration" among individuals enables agents to create global patterns of collective behavior that allow every individual agent to achieve its adaptive aims (Kennedy and Eberhart 2001). Based on our previous results and in order to illustrate the ACACIA features, we tried to find out whether setting SMC-III on also compensated for the disadvantage of having a narrower neighborhood angle.

## Method

We performed a series of simulations. For each simulation, we set one goal and 50 randomly distributed agents, with random headings and a neighborhood radius of 10 patches and a scan resolution of 10 scan lines. We systematically varied the independent variables according to a three-factor design: 2 (SMC-III on or off) $\times$ 2 (neighborhood angle 120°

or 180°) × 2 (0 or 5 obstacles). One sixty independent simulations were run for each design cell, thus there were 1,280 simulations in all. We measured the percentage of agents that reached the goal after 400 simulation steps as a dependent variable.

## Results and discussion

An analysis of variance was performed on the percentage of agents that reached the goal. The results showed statistically significant effects for the three main factors: (a) when the neighborhood angle was set to 180°, a greater percentage of agents reached the goal than when it was set to 120° ($M = 40.32$ and 34.54, respectively; $F_{1,1272} = 45.45$, $P < 0.0001$); (b) a higher percentage of agents reached the goal when SMC-III was on than when it was off ($M = 41.84$ and 33.02, respectively; $F_{1,1272} = 105.78$, $P < 0.0001$); and (c) the presence of obstacles decreased the percentage of agents reaching the goal, compared with when there were no obstacles ($M = 32.01$ and 42.85, respectively; $F_{1,1272} = 159.78$, $P < 0.0001$). The analysis of variance also indicated statistically significant effects between neighborhood angle and number of obstacles ($F_{1,1272} = 5.20$, $P < 0.05$), but not between neighborhood angle and SMC-III, between SMC-III and number of obstacles, and between the three factors.

The results show that a neighborhood angle of 180° increased the agents' chances of reaching the goal, even when the complexity of the environment was increased to five obstacles. When SMC-III was set on results in an increase of the number of the agents reaching the goal. Nevertheless, contrarily to what we expected, the perceptual disadvantage of the agents with a neighborhood angle of 120° was not compensated by setting SMC-III on. However, some results of this simulation experiment confirmed previous findings about SMC-III (Miñano and Beltran 2004).

## Final comments

We have shown that in some cases it is possible to reach a long-term goal through the collective behavior that emerges from a set of sensorial-motor couplings, and it is not necessary for the agent to generate an overall representation of its environment. Thus, self-organized cognition based on a set of sensorial-motor couplings could show a promising way to implement complex behavior and reasoning. Therefore, in a future version, in order to improve the performance of the ACACIA agents, they should build on their knowledge through learning (as they would be initially naïve about their environment); it could be made possible by implementing a learning-classification system in each agent (Holland 1995). Other features to be included in future versions are individual differences in the agents' learning and perception, perception errors (e.g., agents could mistake goals for obstacles) and inter-agent communication.

## Availability

ACACIA can be downloaded from http://www.ub.es/comporta/gcai.htm. To run it on Windows, Java Runtime Environment and StarLogo 2.21 must be preinstalled. StarLogo 2.21 can be downloaded from http://education.mit.edu/starlogo/.

**Acknowledgments** This research project was supported by grants from the Directorate General for Research of the Government of Catalonia (2001SGR-00140), the Directorate General for Higher Education and Culture of the Spanish Government (BSO 2001-2844) and the French Institut National de Recherche en Informatique et en Automatique (02N81/0011). This research project was also supported by funds from the European Union's FEDER program. The authors would like to thank two anonymous reviewers for their comments on an earlier version of the manuscript.

## References

Bakker B (2000) The adaptative behavior approach to psychology. Cogn Process 1:39–70

Braitenberg V (1984) Vehicles: experiments in synthetic psychology. MIT Press, Cambridge

Brooks RA (1999) Cambrian intelligence: the early history of the new AI. Bradford Books/MIT Press, Cambridge

Colella V, Klopfer E, Resnick M (2001) Adventures in modeling. Exploring complex, dynamic systems with StarLogo. Teachers College Press, New York

Couzin ID, Krause J, James R, Ruxton GD, Franks NR (2002) Collective memory and spatial sorting in animal groups. J Theor Biol 218:1–11. doi:10.1006/jtbi.2002.3065

Epstein JM, Axtell R (1996) Growing artificial societies: social science from the bottom-up. MIT Press, Washington

Hemelrijk CK (1996) Dominance interactions, spatial dynamics and emergent reciprocity in a virtual world. In: Maes P, Mataric M, Meyer JA, Pollack J, Wilson SW (eds) From animals to animats: proceedings of the fourth international conference on simulation of adaptive behavior. MIT Press, Cambridge, pp 545–552

Hemelrijk CK (2003) Female co-dominance in a virtual world: ecological, cognitive, social and sexual causes. Behavior 140:1247–1273. doi:10.1163/156853903771980585

Holland JH (1995) Hidden order: how adaptation builds complexity. Perseus Books, Reading

Kennedy J, Eberhart RC (2001) Swarm intelligence. Morgan Kaufmann, San Francisco

Maes P (1997) Modeling adaptive autonomous agents. In: Langton CG (ed) Artificial life: an overview. MIT Press, Cambridge, pp 135–162

Maris M, te Boekhorst R (1996) Exploiting physical constraints: heap formation through behavioral error in a group of robots. In: Asada M (ed) Proceedings of the international conference on intelligent robots and systems. Senri Life Science Center, Osaka, pp 1655–1660

Meyer JA, Guillot A (1991) Simulation of adaptive behavior in animats: review and prospect. In: Meyer JA, Wilson SW (eds) From animals to animats 1: proceedings of the first international conference on simulation of adaptive behavior. Bradford Books/ MIT Press, Cambridge, pp 2–14

Miñano M, Beltran FS (2004) Reaching long-term goals based on local interaction between the organism and its environment: computer simulations based on adaptive behavior. Percept Mot Skills 99:27–33

Miñano M, Zibetti E (2005) Reaching long-term goals based on local categorization. Unpublished manuscript, Université de Paris-8

Resnick M (1994) Turtles, termites and traffic jams: explorations in massively parallel microworlds. MIT Press, Cambridge

Reynolds CW (1993) An evolved, vision-based behavioral model of coordinated group motion. In: Meyer JA, Roitblat HL, Wilson SW (eds) From animals to animats 2: proceedings of the second international conference on simulation of adaptive behavior. MIT Press, Cambridge, pp 384–392

Schreckenberg M, Sharma SD (eds) (2002) Pedestrian and evacuation dynamics. Springer, New York

Werner GM, Dyer MG (1992) Evolution of herding behaviour in artificial animals. In: Meyer JA, Roitblat HL, Wilson SW (eds) From animals to animats 2: proceedings of the second international conference on simulation of adaptive behaviour. MIT Press, Cambridge, pp 393–399

Zibetti E, Quera V, Beltran FS, Tijus C (2001a) Contextual categorization: a mechanism linking perception and knowledge in modelling and simulating perceived events as actions. In: Akman V, Bouquet P, Thomason R, Young RA (eds) Modeling and using context. Springer, Berlin, pp 395–408

Zibetti E, Quera V, Tijus C, Beltran FS (2001b) Reasoning based on categorisation for interpreting and acting: a first approach. Mind Soc 4(2):89–106